

**Выполнить задания 2.1-2.4 отчет содержащий скрины прикрепить на уч. портал**

#### 1. Теоретические сведения

Во время проектирования базы данных необходимо заботиться о целостности данных. Правильная структура таблиц позволяет защитить данные от нарушения связей и внесения неверных значений. Вы должны определить наилучший путь обеспечения целостности данных. Целостность данных основывается на стойкости и точности данных, которые хранит база данных.

Существуют различные типы целостности данных:

- **Целостность полей** – указывает набор значений данных, которые являются правильными для поля, и определяет, возможно ли использование нулевого значения. Целостность полей часто всего (и лучше) обеспечивается с помощью ограничения CHECK, формата (с помощью шаблона) или региона возможных значений для поля.
- **Целостность таблицы** – требуют, чтобы все строки в таблице имели уникальный идентификатор, называемый **первичным ключом**. В некоторых случаях можно разрешить удаление записей, но чаще всего оно должно быть запрещено.
- **Целостность ссылок** – подразумевает отношения между первичным ключом (таблицы, на которую ссылаются) и внешним ключом (таблицы, которая ссылается на другую) всегда защищенными. **Строка основной таблицы, на которую ссылаются, не может быть удалена и первичный ключ не может быть изменен, если вторичный ключ ссылается на строку, пока не будет уничтожена связь.** Иначе связь нарушается и восстановить ее потом становится проблематичным.

Такая целостность обеспечивается с помощью ограничений CHECK, DEFAULT и внешнего ключа.

**Ограничение** – это основной метод обеспечения целостности данных.

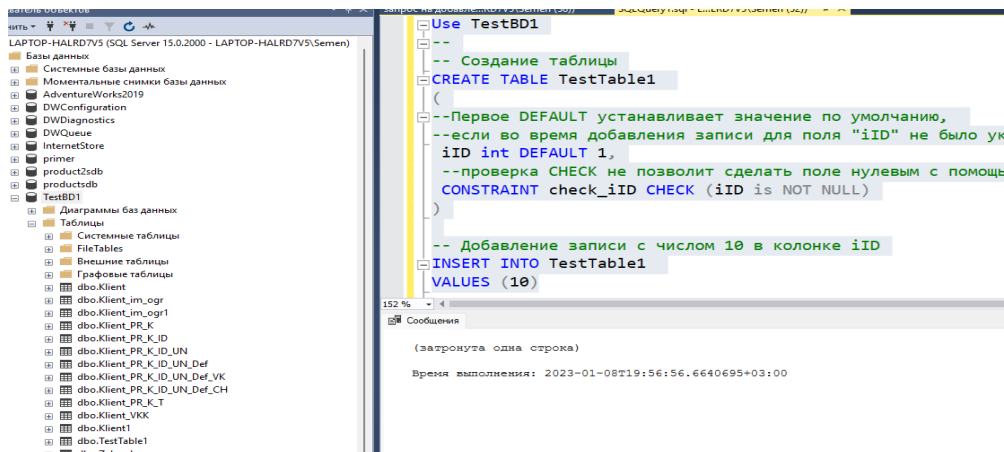
Как мы уже знаем, ограничения могут создаваться во время создания таблицы (CREATE TABLE) или редактирования (ALTER TABLE). Если ограничение назначается отдельному полю, оно называется ограничением уровня поля. Если ограничение ссылается на несколько полей, оно называется ограничением уровня таблицы, даже если оно ссылается не на все колонки таблицы. **Имена ограничений** должны быть уникальными для базы данных. Если не указывать имена, то сервер сгенерирует значение самостоятельно.

**Теперь рассмотрим уже знакомые нам операторы ограничений, только с точки зрения обеспечения целостности данных.**

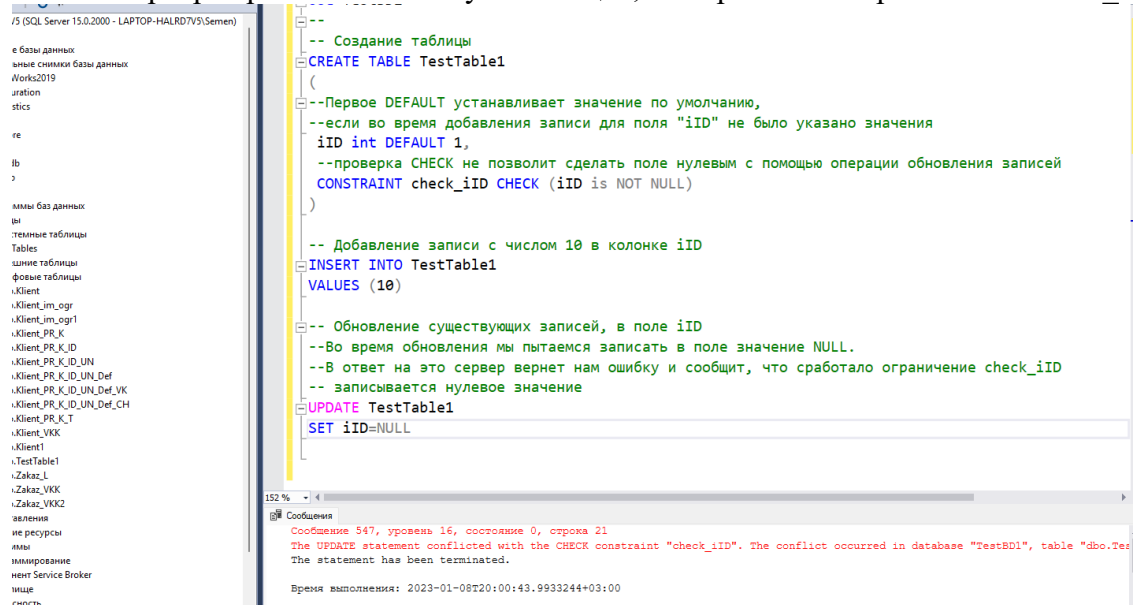
#### 2. Практическая часть

**!!Примечание:** работаете в своей БД здесь это **TestDBD1**, у вас это может быть другая

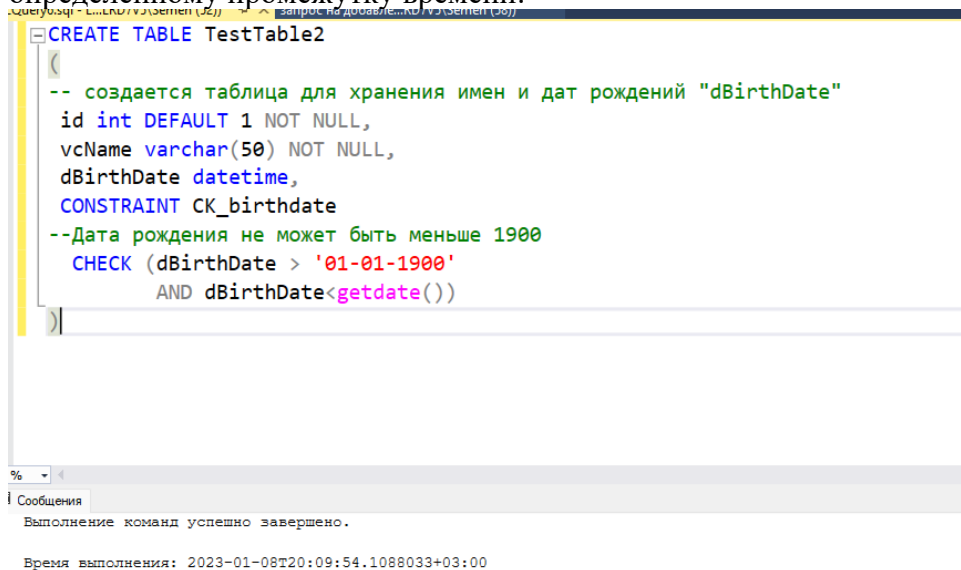
**Задание 2.1 Ограничение DEFAULT** помещает значение в колонку, когда оно не было указано в операторе INSERT.



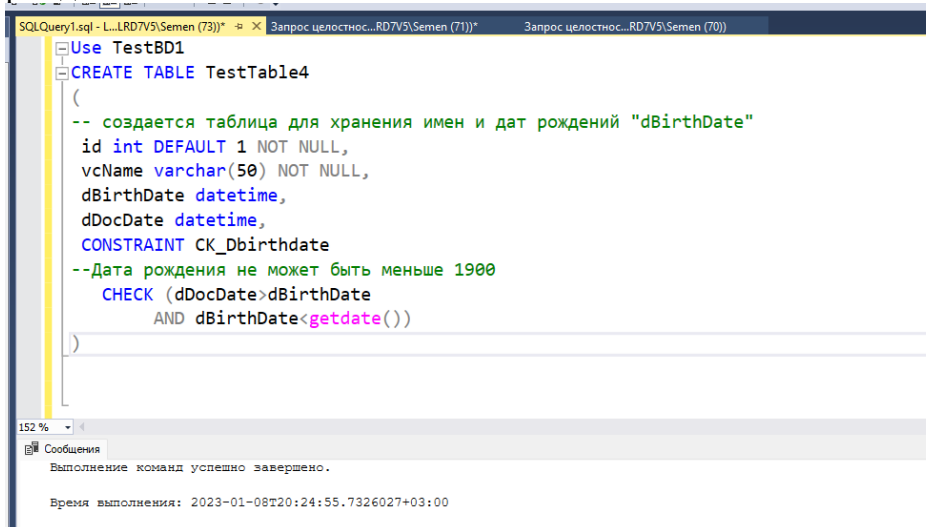
**Задание 2.2** Если во время обновления мы пытаемся записать в поле значение NULL. В ответ на это сервер вернет нам ошибку и сообщит, что сработало ограничение check\_iID



**Задание 2.3** Ограничение CHECK ограничивает данные, которые пользователь может ввести в определенную колонку указанными значениями. Следующий пример добавляет ограничение, чтобы гарантировать, что день рождения соответствует определенному промежутку времени:



**Задание 2.4** В следующей таблицы помимо даты рождения человека в таблице будет храниться дата выдачи паспорта – "dDocDate". Вполне логично, что дата выдачи паспорта должна быть больше даты рождения и меньше текущей. Паспорт не может быть выдан до рождения, поэтому в следующей таблице мы гарантируем целостность поля даты рождения:



```
SQLQuery1.sql - L:\RD7V5\Semem (73)) *  Запрос целостнос...RD7V5\Semem (71)) *  Запрос целостнос...RD7V5\Semem (70))
Use TestBD1
CREATE TABLE TestTable4
(
    -- создается таблица для хранения имен и дат рождений "dBirthDate"
    id int DEFAULT 1 NOT NULL,
    vcName varchar(50) NOT NULL,
    dBirthDate datetime,
    dDocDate datetime,
    CONSTRAINT CK_Dbirthdate
    --Дата рождения не может быть меньше 1900
    CHECK (dDocDate>dBirthDate
        AND dBirthDate<getdate())
)
```

Сообщения  
Выполнение команд успешно завершено.  
Время выполнения: 2023-01-08T20:24:55.7326027+03:00

## Задание 2.5 Ключи

Ограничение PRIMARY KEY определяет первичный ключ таблицы, который уникально идентифицирует строку. Это гарантирует целостность таблицы. Когда мы изучали оператор PRIMARY KEY, то свойства первичного ключа:

- в таблицы может быть только один первичный ключ, но этот ключ может состоять из нескольких полей;
- поле не может содержать нулевого значения;
- для полей первичного ключа создается индекс, который может быть кластерным или не кластерным;
- Не смотря на то, что для полей первичного ключа не создается ограничения уникальности, оно подразумевается и две записи не могут содержать одинаковых значений в первичном ключе.

Ограничение FOREIGN KEY (внешний ключ) гарантирует ссылочную целостность. Ограничение внешнего ключа определяет ссылку на колонку с первичным ключом или уникальную колонку в этой же или другой таблице. С помощью такого ключа обеспечивается целостность связей между таблицами.

**Связываться можно не только с первичным ключом, но и с полем, содержащим ограничение уникальности.**

Ограничение внешнего ключа включает опцию CASCADE, которая позволяет любые изменения сделанные в уникальной колонке или первичном ключе автоматически переносить в значение внешнего ключа. Такое действие называется целостностью каскадных ссылок.

Опция REFERENCE команд CREATE TABLE и ALTER TABLE поддерживают опции ON DELETE и ON UPDATE. Эти опции позволят вам указать опции CASCADE и NO ACTION:

[ ON DELETE { CASCADE | NO ACTION } ]

[ ON UPDATE { CASCADE | NO ACTION } ]

**NO ACTION указывает что любые попытки удалить или обновить ключ, на который ссылается вторичный ключ в другой таблице заканчиваются ошибкой, и изменения откатываются. Это значение по умолчанию и без особой надобности не стоит включать каскадных действий.**

**Уникальность/ Ограничение UNIQUE (уникальность)** указывает, что две строки в колонке не могут содержать одно и тоже значение. Это ограничение обеспечивает целостность таблицы с уникальным индексом. **Ограничение уникальности эффективно, когда вы уже имеете первичный ключ, но хотите гарантировать, что другое поле тоже уникально.**

В отличие от первичного ключа, у уникального поля может быть одна строка с нулевым значением. Это необходимо учитывать и если вы хотите, чтобы поле не могло содержать нулевого значения, добавьте ограничение CHECK, например:

```
CREATE TABLE TestTable5
(
    --создается два ограничения на поле "iID": одно на уникальность и одно на запрет NULL значений
    iID int DEFAULT 1,
    CONSTRAINT check_iID2 CHECK (iID is NOT NULL),
    CONSTRAINT u_iID
        UNIQUE NONCLUSTERED(iID)
)
```

Сообщения

Выполнение команд успешно завершено.

Время выполнения: 2023-01-08T21:08:29.1626302+03:00

В примере создается два ограничения на поле "iID": одно на уникальность и одно на запрет NULL значений.

Ограничений уникальности может быть несколько в таблице, и для каждого такого поля будет создаваться индекс.

### **Отключение ограничений**

Для повышения производительности, иногда разумно отключить ограничения. Для примера, более эффективно позволить выполнить большую операцию обновления или вставки данных, без ограничений.

Когда вы определяете ограничение на таблицу, которая уже содержит данные, MS SQL Server проверяет данные автоматически, гарантируя, что после создания ограничения, существующие данные соответствуют требованиям.

**Отключать можно только ограничения CHECK и FOREIGN KEY. Другие ограничения должны быть удалены и потом снова добавлены.**

Для отключения проверки, когда вы добавляете ограничения CHECK и FOREIGN KEY на таблицу с существующими данными, включите опцию **WITH NOCHECK** в оператор **ALTER TABLE**.

В следующем примере, мы добавляем ограничение **FOREING KEY**. Ограничение не проверяет существующие данные на момент добавления ограничения:

```
ALTER TABLE TestTable
WITH NOCHECK
ADD CONSTRAINT FK_TestTable
FOREIGN KEY (Field1)
REFERENCES PrimaryTable(Field2)
```

Вы можете отключить проверку ограничений на существующие ограничения CHECK и FOREIGN KEY так, что любые данные, которые вы изменяете или добавляете в таблицу, не проверяются с ограничением.

Вы можете захотеть отключить проверку ограничений когда:  
вы уже убедились, что данные соответствуют требованиям;  
вы хотите загрузить данные, которые не соответствуют ограничению.

Включение ограничения, которое было отключено, требует выполнения другого оператора **ALTER TABLE**, которое содержит опцию **CHECK** или **CHECK ALL**.

```
ALTER TABLE имя таблицы
{CHECK | NOCHECK} CONSTRAINT
{ALL | ограничение [, ...]}
```