

Частное учреждение образования
«Колледж бизнеса и права»

УТВЕРЖДАЮ
Ведущий методист
колледжа
_____ Е.В. Паскал
«___» _____ 2022

Специальность: «Программное обеспечение информационных технологий»	Учебная дисциплина: «Базы данных и системы управления базами данных»
--	--

ЛАБОРАТОРНАЯ РАБОТА № 9

Инструкционно-технологическая карта

Тема: Создание таблиц. Создание связей между таблицами.

Цель работы: научиться создавать таблицы базы данных. Уметь объявлять декларативные ограничения целостности при использовании команды создания таблиц. Научиться устанавливать связи между таблицами.

Время выполнения: 2 часа

Содержание работы

1. Теоретические сведения для выполнения работы
2. Порядок выполнения работы
3. Пример выполнения работы
4. Контрольные вопросы
5. Литература

1. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ ДЛЯ ВЫПОЛНЕНИЯ РАБОТЫ

Для создания таблиц применяется команда CREATE TABLE. С этой командой можно использовать ряд операторов, которые определяют столбцы таблицы и их атрибуты. И кроме того, можно использовать ряд операторов, которые определяют свойства таблицы в целом. Одна база данных может содержать до 2 миллиардов таблиц.

Общий синтаксис создания таблицы выглядит следующим образом:

```
CREATE TABLE название_таблицы  
(название_столбца1 тип_данных атрибуты_столбца1,  
название_столбца2 тип_данных атрибуты_столбца2,  
.....  
название_столбцаN тип_данных атрибуты_столбцаN,  
атрибуты_таблицы  
)
```

В самом простом виде команда CREATE TABLE должна содержать как минимум имя таблицы, имена и типы столбцов.

Таблица может содержать от 1 до 1024 столбцов. Каждый столбец должен иметь уникальное в рамках текущей таблицы имя, и ему должен быть назначен тип данных.

Ограничения могут создаваться во время создания таблицы (CREATE 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, но сгенерированное значение слишком сложное и не отражает сути происходящей проверки. Поэтому, рекомендуется всегда указывать имена самостоятельно и так, чтобы оно отражало суть происходящей проверки и таблицы, в которой происходит проверка.

SQL-сервер поддерживает следующие ограничения:

NOT NULL - проверка на непустое значение. **NULL** - специальное понятие в СУБД, которое означает "пусто". "Пусто" и "0(ноль)" не равны друг другу!

UNIQUE - проверка на уникальность. Вставляемое значение должно быть уникально для данного столбца по всей таблице. Может содержать пустые значения.

PRIMARY KEY - первичный ключ. Значение в столбце считается первичным ключом, если оно непустое и уникально в пределах столбца данной таблицы. Первичный ключ может быть составным и представлять собой комбинацию столбцов. Тогда чтобы считаться первичным ключом, каждое из группы значений не должно быть пустыми и формируемые строки значений первичного ключа должны быть уникальны в пределах таблицы.

FOREIGN KEY - внешний ключ. Назначает столбец или комбинацию столбцов в текущей (родительской) таблице в качестве внешнего ключа для ссылки из других таблиц. **REFERENCES** - указатель ссылки (или родительский ключ). Указывает на столбец (комбинацию столбцов) в родительской таблице, ограничивающую значения в текущей (дочерней) таблице.

CHECK - проверка фиксированного условия. В данном ограничителе явно указывается условие, которое должно выполняться для вставляемого или модифицируемого значения в столбце. При формировании условий с некоторыми ограничениями могут использоваться функции, но CHECK не может содержать подзапросы (SELECT).

2. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

1. Изучить теоретическую часть настоящей инструкционно-технологической карты.

2. Рассмотреть создание таблиц баз данных и объявление декларативных ограничений целостности, описанные в разделе «Пример выполнения работы» настоящей инструкционно-технологической карты.

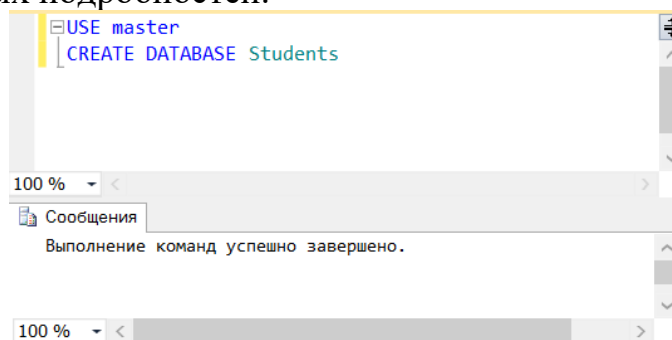
3. Получить у преподавателя индивидуальное задание и выполнить лабораторную работу в соответствии с вариантом задания согласно описанной в разделе «Пример выполнения работы» методике настоящей инструкционно-технологической карты.

4. Ответить на контрольные вопросы.

3. ПРИМЕР ВЫПОЛНЕНИЯ РАБОТЫ

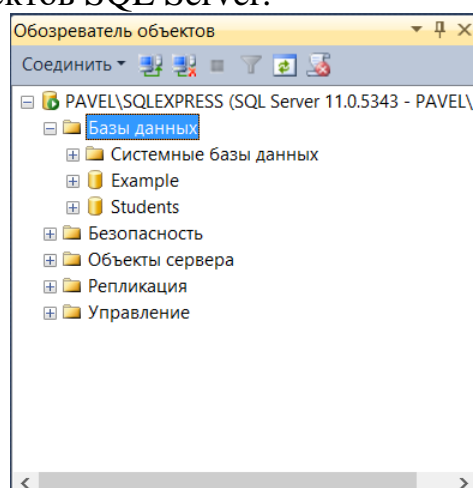
1. Создание базы данных

В следующем примере показан код для создания простой базы данных, без указания дополнительных подробностей.



Такая сокращенная форма инструкции CREATE DATABASE возможна благодаря тому, что почти все ее параметры имеют значения по умолчанию. По умолчанию система создает два файла. Файл данных имеет логическое имя Students и исходный размер 2 Мбайта. А файл журнала транзакций имеет логическое имя Students_log и исходный размер 1 Мбайт.

Результат представлен на рисунке ниже. Файл базы данных Students появился в обозревателе объектов SQL Server.



2. Присоединение и отсоединение базы данных

В следующем примере происходит отсоединение базы данных Students с помощью системной процедуры **sp_detach_db**, которая имеет один файл Students.mdf и один журнал Students.ldf.

```
USE master
go
sp_detach_db 'Students'
```

Сообщения
Выполнение команд успешно завершено.

В следующем примере снова необходимо прикрепить базу данных Students для дальнейшей работы с помощью системной процедуры **sp_attach_db**. Код представлен ниже.

```
use master
go
sp_attach_db 'Students', 'C:\Program Files (x86)\Microsoft SQL Server\MSSQL11.SQLEXPRESS\MSSQL\DATA\Students.mdf',
'C:\Program Files (x86)\Microsoft SQL Server\MSSQL11.SQLEXPRESS\MSSQL\DATA\Students_log.ldf'
go
```

Сообщения
Выполнение команд успешно завершено.

Если база данных, которую необходимо прикрепить, имеет дополнительные файлы данных или журналы, их указывают все через запятую в хранимой процедуре **sp_attach_db**.

3. Создание таблиц базы данных

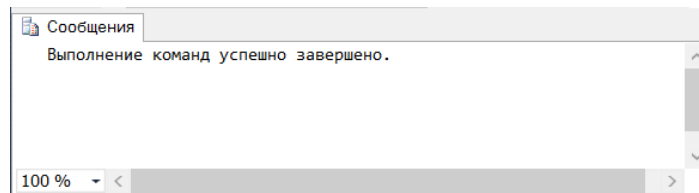
В следующем примере показан код создания таблиц базы данных Students:

таблицы Специальности, таблица Предметы, таблица Оценки, таблица Студенты. Если необходимо создать вычисляемое поле, то в команде Create Table у вычисляемого поля вместо типа данных после команды AS нужно указать выражение для вычисления значения. Пример создания вычисляемого поля Средний балл показан при создании таблицы Оценки

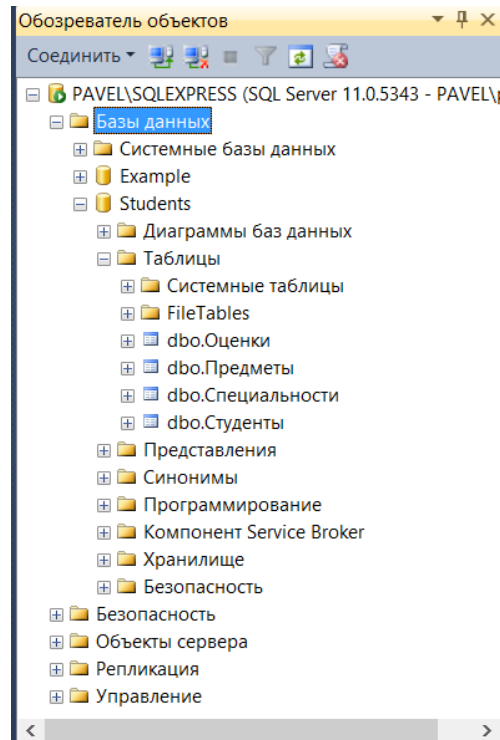
```
USE Students
CREATE TABLE Специальности --Создание таблицы Специальности
([Код специальности] bigint identity (1,1) not null,
[Наименование специальности] varchar(50),
[Описание специальности] varchar(max))
CREATE TABLE Предметы --Создание таблицы Предметы
([Код предмета] bigint identity(1,1) not null,
[Наименование предмета] varchar (50),
[Описание предмета] varchar(max))
CREATE TABLE Оценки --Создание таблицы Оценки
([Код студента] bigint,
[Дата экзамена 1] datetime,
[Код предмета 1] bigint,
[Оценка 1] tinyint,
[Дата экзамена 2] datetime,
[Код предмета 2] bigint,
[Оценка 2] tinyint,
[Дата экзамена 3] datetime,
[Код предмета 3] bigint,
[Оценка 3] tinyint,
[Средний балл] as ([Оценка 1]+[Оценка 2]+[Оценка 3])/3)
CREATE TABLE Студенты --Создание таблицы Студенты
([Код студента] bigint identity NOT NULL,
ФИО VARCHAR (50),
Пол VARCHAR (10),
[Дата рождения] DATETIME,
Родители VARCHAR (50),
Адрес VARCHAR(MAX),
Телефон VARCHAR(15),
[Паспортные данные] VARCHAR(MAX),
[Номер зачетки] BIGINT,
[Дата поступления] DATETIME,
Группа VARCHAR (10),
Курс TINYINT,
[Код специальности] BIGINT,
[Очная форма обучения] BIT)
```

100 %

В случае успешного выполнения кода системы выдаст следующее сообщение:

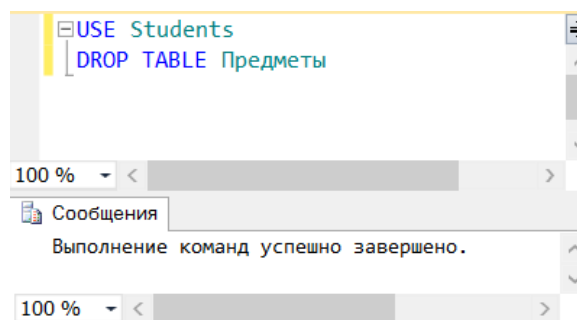


В результате в обозревателе объектов появятся таблицы базы данных Students:

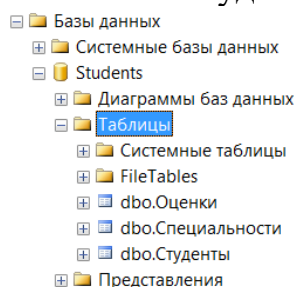


4. Создание потенциальных ключей

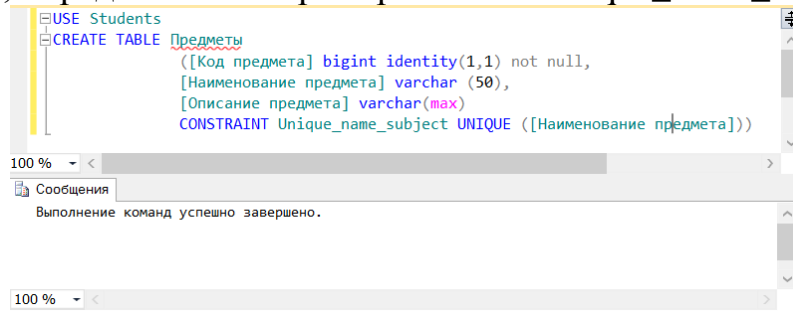
Прежде чем выполнять следующие примеры необходимо удалить таблицу Предметы используя инструкцию DROP TABLE Предметы, как показано на рисунке ниже.



В результате обозреватель объектов будет выглядеть следующим образом:



Каждое значение столбца Наименование предмета таблицы Предметы является уникальным, включая значения NULL. В следующем примере показано применение предложения UNIQUE к столбцу Наименование предмета. Явное имя ограничения, определяемое в примере ниже – Unique_name_subject.



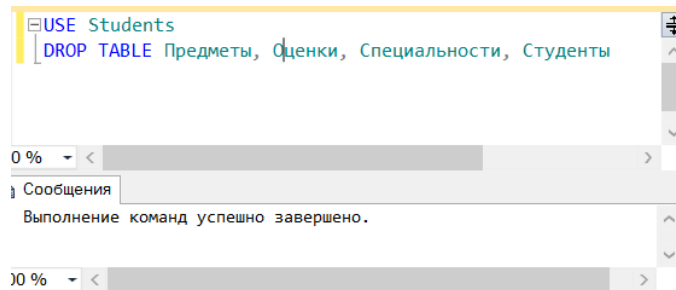
```
USE Students
CREATE TABLE Предметы
(
    ([Код предмета] bigint identity(1,1) not null,
    [Наименование предмета] varchar (50),
    [Описание предмета] varchar(max)
    CONSTRAINT Unique_name_subject UNIQUE ([Наименование предмета]))

```

Сообщения
Выполнение команд успешно завершено.

5. Создание первичных ключей

Прежде чем выполнять следующие примеры необходимо удалить все таблицы базы данных Students используя инструкцию DROP TABLE, как показано на рисунке ниже.



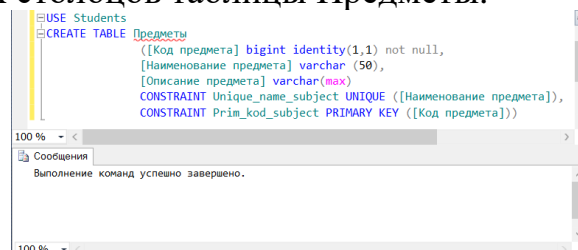
```
USE Students
DROP TABLE Предметы, Оценки, Специальности, Студенты

```

Сообщения
Выполнение команд успешно завершено.

Первичный ключ таблиц базы данных Students определяется посредством декларативного ограничения для обеспечения целостности с определенным именем для соответствующей таблицы.

В примере ниже, создается таблица Предметы, в которой определен первичный ключ под именем prim_kod_subject. Это ограничение для обеспечения целостности является ограничением уровня таблицы, поскольку оно указывается после определения всех столбцов таблицы Предметы.

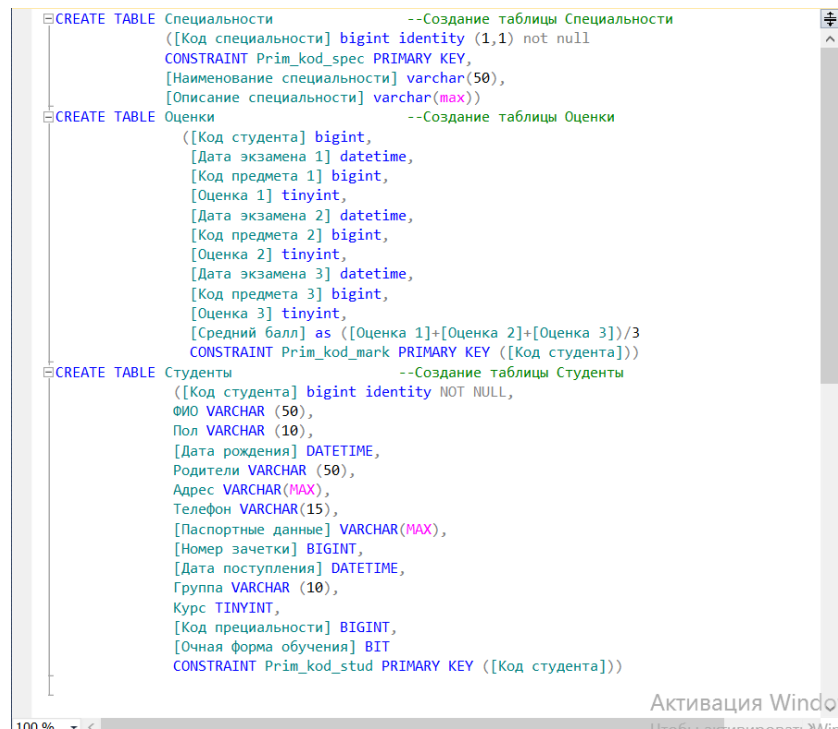


```
USE Students
CREATE TABLE Предметы
(
    ([Код предмета] bigint identity(1,1) not null,
    [Наименование предмета] varchar (50),
    [Описание предмета] varchar(max)
    CONSTRAINT Unique_name_subject UNIQUE ([Наименование предмета]),
    CONSTRAINT Prim_kod_subject PRIMARY KEY ([Код предмета]))

```

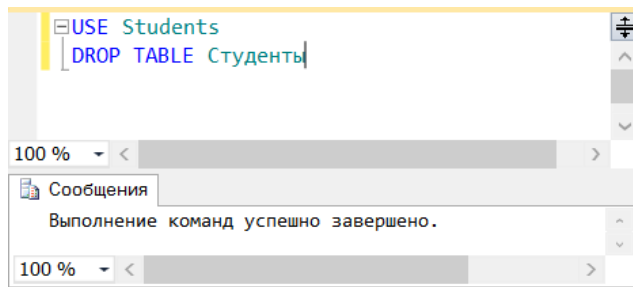
Сообщения
Выполнение команд успешно завершено.

В результате выполнения кода, представленного на рисунке ниже, аналогично созданию таблицы Предметы создаются оставшиеся таблицы базы данных Students: Оценки, Студенты, Специальности. При создании таблицы Специальности предложение PRIMARY KEY принадлежит к объявлению столбца Код специальности, наряду с объявлением его типа данных и свойства содержать значения NULL. По этой причине это ограничение называется *ограничением на уровне столбца*.

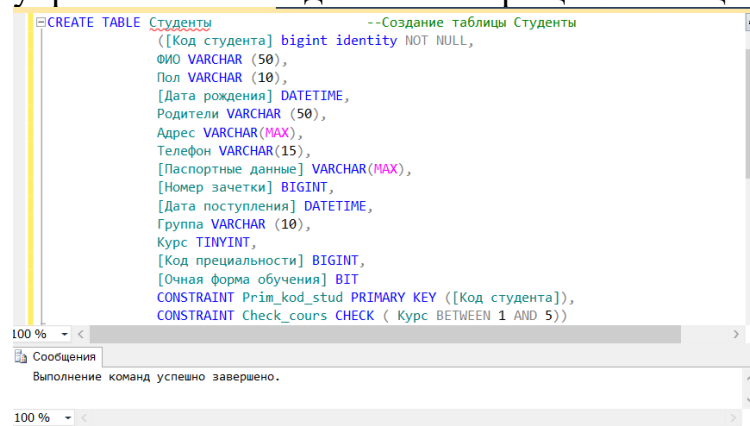


6. Создание проверочных ограничений

Прежде чем выполнять следующие примеры необходимо удалить таблицу Студенты, используя инструкцию DROP TABLE Студенты, как показано на рисунке ниже.

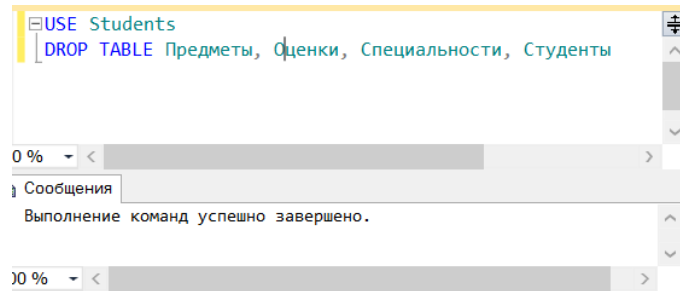


В результате выполнения кода, представленного на рисунке ниже, создается таблица Студенты, включающая столбец Курс, содержащий проверочное ограничение Check_cours: курс может принимать значения от 1 до 5. При вставке нового значения, отличающегося от значений в диапазоне от 1 до 5, или при попытке изменения существующего значения на значение, отличающегося от этих значений, система управления базой данных возвращает сообщение об ошибке.

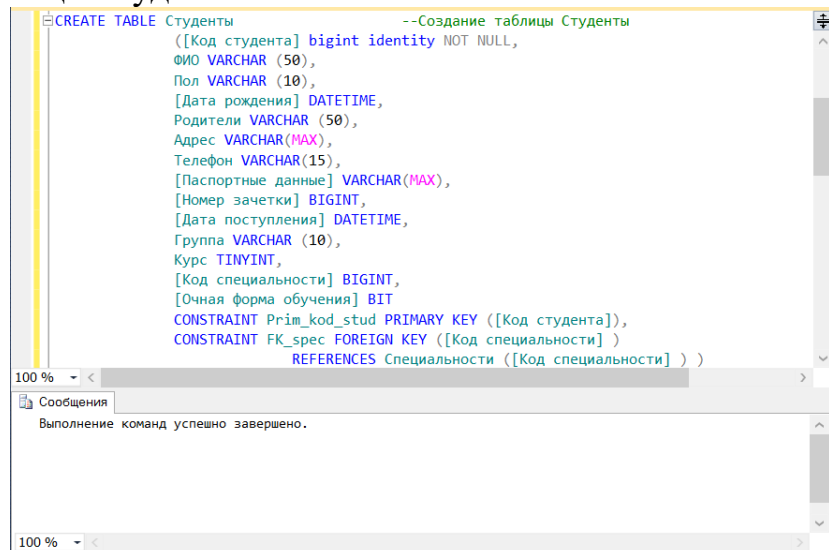


7. Создание внешних ключей

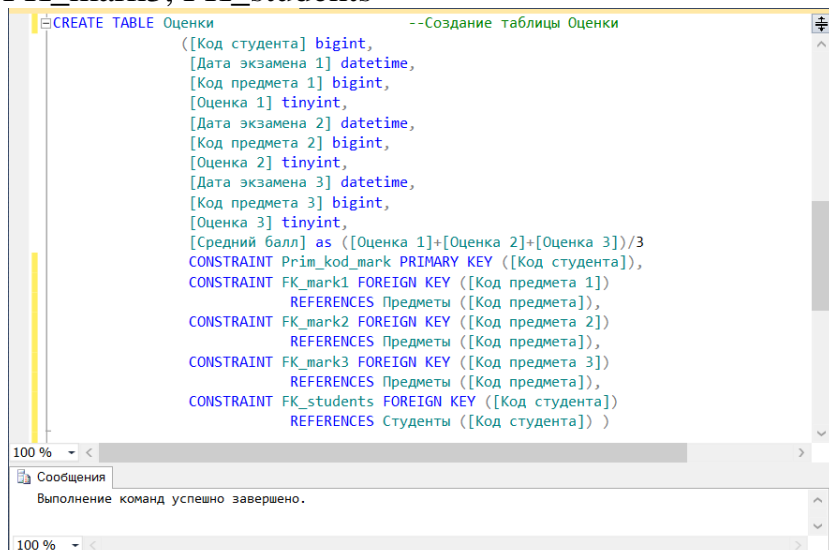
Прежде чем выполнять следующие примеры необходимо удалить все таблицы базы данных Students используя инструкцию DROP TABLE, как показано на рисунке ниже.



В примере ниже, создается таблица Студенты, в которой определены два декларативные ограничения для обеспечения целостности: Prim_kod_stud и FK_spec. Оба ограничения являются уровня таблицы, где первое указывает первичный ключ, а второе — внешний ключ таблицы Студенты. Кроме этого, ограничение FK_spec определяет таблицу Специальности как ссылочную таблицу, а ее столбец Код специальности, как соответствующий внешний ключ столбца с таким же именем в таблице Студенты.



Аналогичным образом определяются внешние ключи в таблице Оценки: FK_mark1, FK_mark2, FK_mark3, FK_students



Таблицы Предметы и Специальности не требуют определения декларативные ограничения для обеспечения целостности в виде внешних ключей. Они являются ссылочными таблицами.

8. Создание диаграммы базы данных.

Для создания диаграммы базы данных необходимо выбрать в обозревателе объектов узел созданной базы данных, перейти к пункту Диаграммы баз данных и через ПКМ выбрать Создать диаграмму базы данных. Далее выбрать все таблицы.

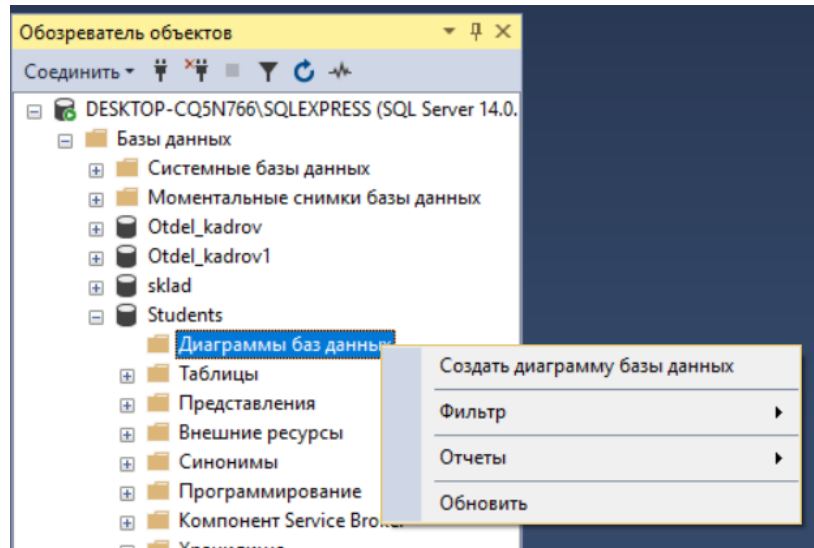
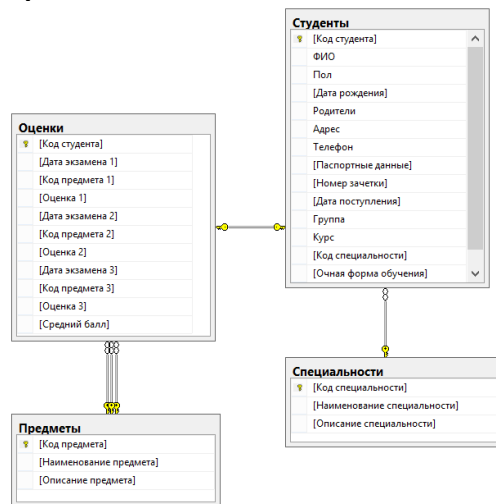


Диаграмма базы данных Students после создания внешних ключей будет выглядеть следующим образом:



9. Применение опций ON DELETE и ON UPDATE

Прежде чем выполнять следующие примеры необходимо удалить таблицу Студенты, используя инструкцию DROP TABLE Студенты.

Использование опций ON DELETE и ON UPDATE показано в следующем примере при создании таблицы Студенты.

```

CREATE TABLE Студенты
(
    [Код студента] bigint identity NOT NULL,
    ФИО VARCHAR (50),
    Пол VARCHAR (10),
    [Дата рождения] DATETIME,
    Родители VARCHAR (50),
    Адрес VARCHAR(MAX),
    Телефон VARCHAR(15),
    [Паспортные данные] VARCHAR(MAX),
    [Номер зачетки] BIGINT,
    [Дата поступления] DATETIME,
    Группа VARCHAR (10),
    Курс TINYINT,
    [Код специальности] BIGINT,
    [Очная форма обучения] BIT
    CONSTRAINT Prim_kod_stud PRIMARY KEY ([Код студента]),
    CONSTRAINT FK_spec FOREIGN KEY ([Код специальности])
    REFERENCES Специальности ([Код специальности] )
    ON DELETE CASCADE
    ON UPDATE CASCADE
)
  
```

4. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Назовите инструкции создания базы данных и создания таблицы.
2. Назовите системные процедуры, которые используются для отсоединения базы данных от сервера и присоединения.
3. Дайте определение понятию «ограничения для обеспечения целостности (integrity constraints)».
4. Назовите типы ограничений для обеспечения целостности, которые предоставляются СУБД.
5. Перечислите инструкции языка DDL, с помощью которых определяются декларативные ограничения целостности.
6. Поясните разницу между декларативными ограничениями уровня столбцов и уровня таблицы.
7. Назовите опцию, посредством которой может быть явно присвоено имя декларативному ограничению.
8. Назовите категории декларативных ограничений.
9. Поясните понятие ссылочной целостности.

5. ЛИТЕРАТУРА

1. Петкович, Д. Microsoft SQL Server 2012. Руководство для начинающих: пер. с английского / Д. Петкович. – СПб.: БХВ-Петербург, 2013. – 816 с.: ил.
2. Создание базы данных [Электронный ресурс]. – Режим доступа: <https://docs.microsoft.com/ru-ru/sql/relational-databases/databases/create-a-database?view=sql-server-ver15>

Преподаватель

К.О.Якимович

Рассмотрено на заседании цикловой
комиссии программного обеспечения
информационных технологий №10
Протокол № __ от «__» _____ 2022
Председатель ЦК В.Ю.Михалевич