

Частное учреждение образования
«Колледж бизнеса и права»

УТВЕРЖДАЮ
Заведующий
методическим кабинетом
_____ Е.В. Фалей
«__» _____ 2017

Специальность: «Программное обеспечение информационных технологий»	Дисциплина: «Базы данных и системы управления базами данных»
Составлена на основании учебной программы, утвержденной директором Колледжа бизнеса и права 30.12.2016	

Лабораторная работа № 14
Инструкционно-технологическая карта

Тема: Создание доменов и таблиц

Цель работы:

- получить навыки работы с инструкциями создания базы данных и создания таблицы: CREATE DATABASE, CREATE TABLE;
- получить навыки присоединения и отсоединения базы данных с помощью системных процедур;
- получить навыки работы с различными категориями декларативных ограничений целостности: DEFAULT, UNIQUE, PRIMARY KEY, CHECK, FOREIGN KEY;
- получить навыки работы с псевдонимными типами данных.

Время выполнения: 2 часа

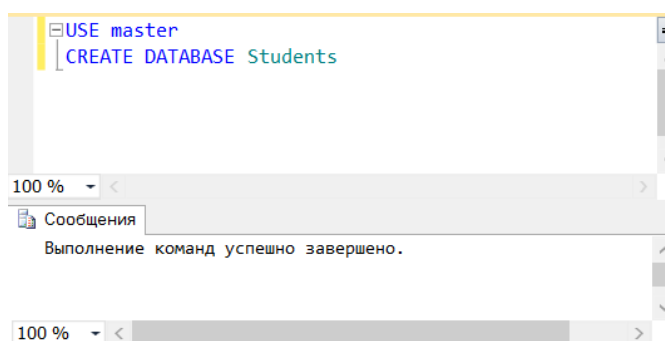
ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

1. Изучить раздел лекционного курса по теме «Описание данных на основе SQL».
2. 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.
3. Ответить на контрольные вопросы.

1. ПРИМЕР ВЫПОЛНЕНИЯ РАБОТЫ

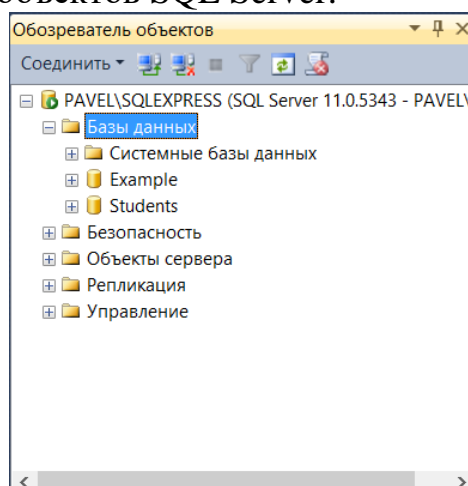
1. Создание базы данных

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



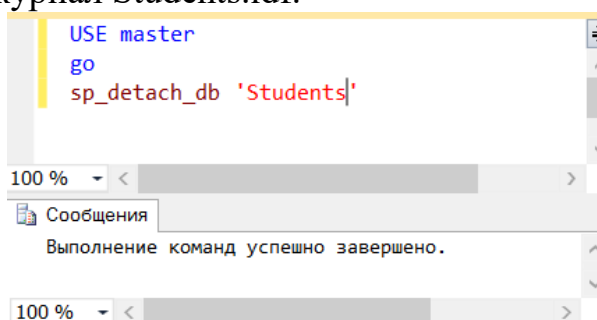
Такая сокращенная форма инструкции CREATE DATABASE возможна благодаря тому, что почти все ее параметры имеют значения по умолчанию. По умолчанию система создает два файла. Файл данных имеет логическое имя Students и исходный размер 2 Мбайта. А файл журнала транзакций имеет логическое имя Students_log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



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

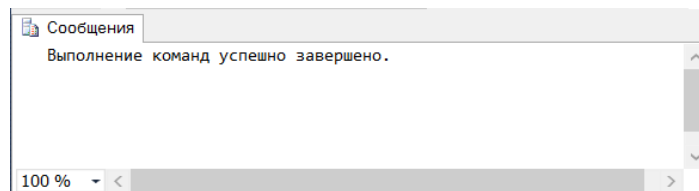
В следующем примере показан код создания таблиц базы данных Students: таблицы Специальности, таблица Предметы, таблица Оценки, таблица Студенты. Если необходимо создать вычисляемое поле, то в команде `Create Table` у вычисляемого поля вместо типа данных после команды `AS` нужно указать выражение для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

```

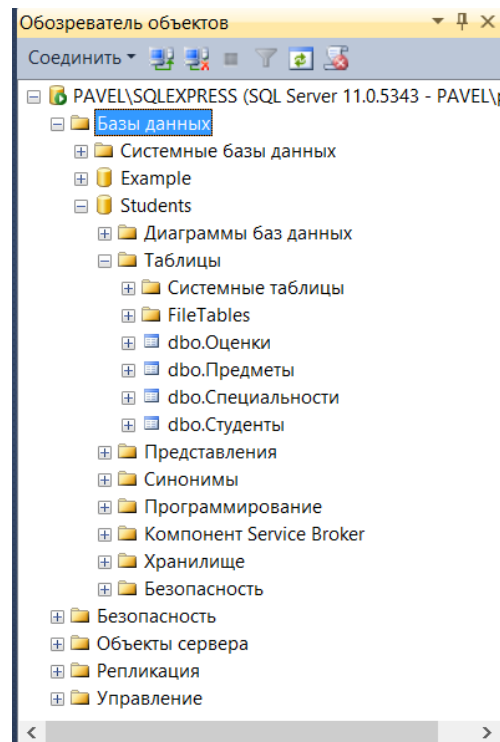
USE Students
CREATE TABLE Специальности --Создание таблицы Специальности
([Код специальности] bigint identity (1,1) not null,
[Наименование специальности] varchar(50),
[Описание специальности] varchar(max))
CREATE TABLE Предметы --Создание таблицы Предметы
([Код предмета] bigint identity(1,1) not null,
[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)

```

В случае успешного выполнения кода системы выдаст следующее сообщение:

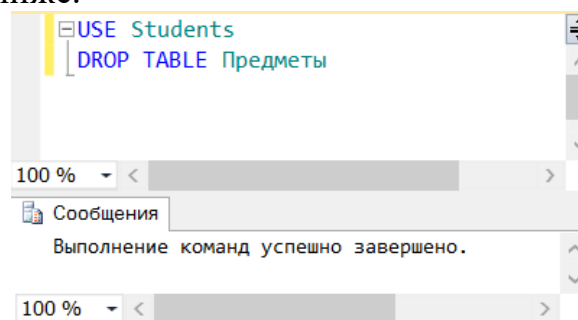


В результате в обозревателе объектов появятся таблицы базы данных Students:

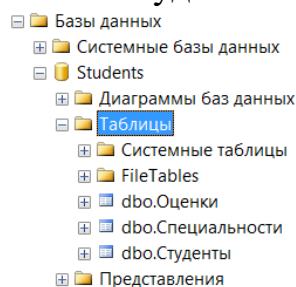


4. Создание потенциальных ключей

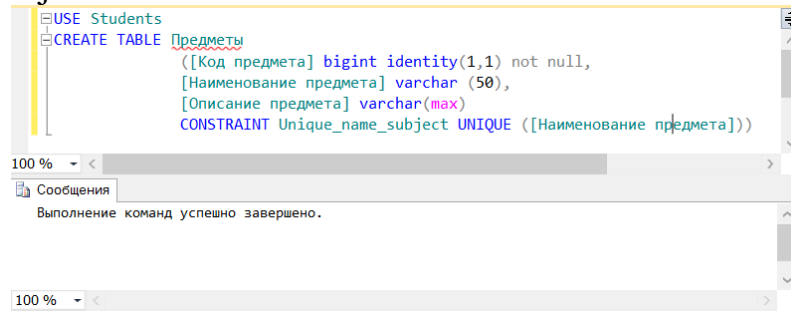
Прежде чем выполнять следующие примеры необходимо удалить таблицу Предметы используя инструкцию DROP TABLE Предметы, как показано на рисунке ниже.



В результате обозреватель объектов будет выглядеть следующим образом:

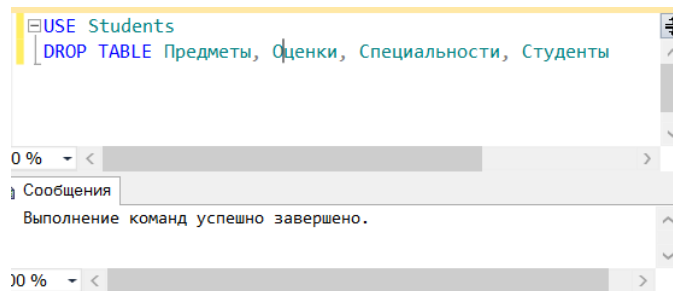


Каждое значение столбца Наименование предмета таблицы Предметы является уникальным, включая значения NULL. В следующем примере показано применение предложения UNIQUE к столбцу Наименование предмета. Явное имя ограничения, определяемое в примере ниже – Unique_name_subject.



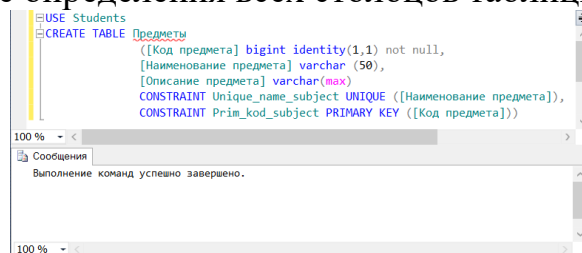
5. Создание первичных ключей

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



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.



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*.

```

CREATE TABLE Специальности --Создание таблицы Специальности
([Код специальности] bigint identity (1,1) not null
CONSTRAINT Prim_kod_spec PRIMARY KEY,
[Наименование специальности] varchar(50),
[Описание специальности] 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
CONSTRAINT Prim_kod_mark PRIMARY KEY ([Код студента]))

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
CONSTRAINT Prim_kod_stud PRIMARY KEY ([Код студента]))

```

6. Создание проверочных ограничений

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

```

USE Students
DROP TABLE Студенты

```

Сообщения
Выполнение команд успешно завершено.

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.

```

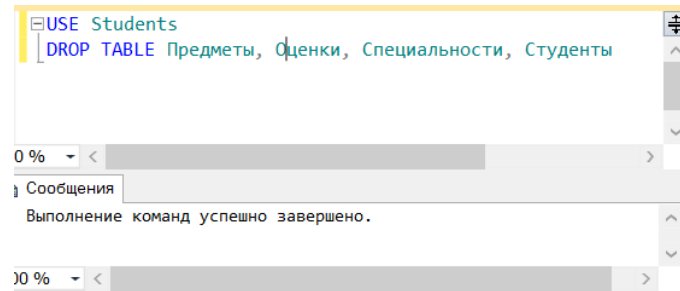
CREATE TABLE Студенты --Создание таблицы Студенты
([Код студента] bigint identity NOT NULL,
ФИО VARCHAR (50),
Пол VARCHAR (10),
[Дата рождения] DATETIME,
Родители VARCHAR (50),
Адрес VARCHAR(MAX),
Телефон VARCHAR(15),
[Паспортные данные] VARCHAR(MAX),
[Номер зачетки] BIGINT,
[Дата поступления] DATETIME,
Группа VARCHAR (10),
Курс TINYINT,
[Код специальности] BIGINT,
[Очная форма обучения] BIT
CONSTRAINT Prim_kod_stud PRIMARY KEY ([Код студента]),
CONSTRAINT Check_cours CHECK ( Курс BETWEEN 1 AND 5))

```

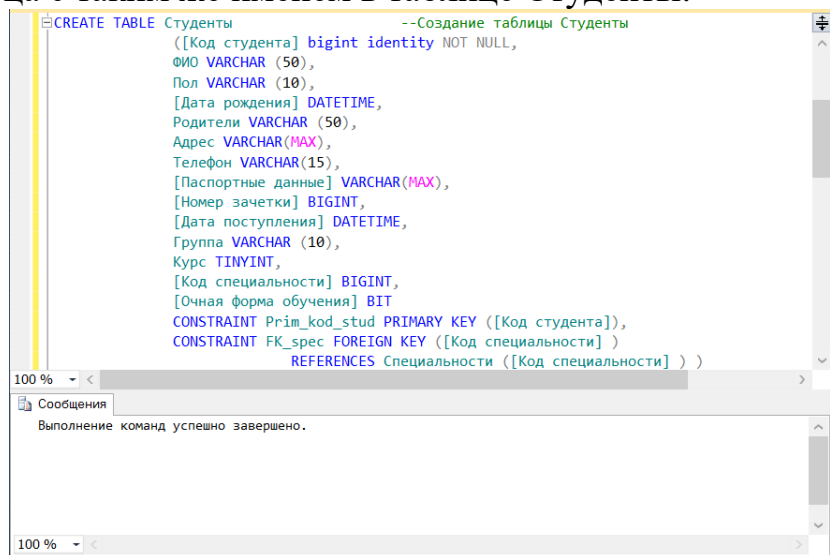
Сообщения
Выполнение команд успешно завершено.

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_spes. Оба ограничения являются уровня таблицы, где первое указывает первичный ключ, а второе — внешний ключ таблицы Студенты. Кроме этого, ограничение FK_spes 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

```

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,
    CONSTRAINT Prim_kod_mark PRIMARY KEY ([Код студента]),
    CONSTRAINT FK_mark1 FOREIGN KEY ([Код предмета 1])
        REFERENCES Предметы ([Код предмета]),
    CONSTRAINT FK_mark2 FOREIGN KEY ([Код предмета 2])
        REFERENCES Предметы ([Код предмета]),
    CONSTRAINT FK_mark3 FOREIGN KEY ([Код предмета 3])
        REFERENCES Предметы ([Код предмета]),
    CONSTRAINT FK_students FOREIGN KEY ([Код студента])
        REFERENCES Студенты ([Код студента])
)

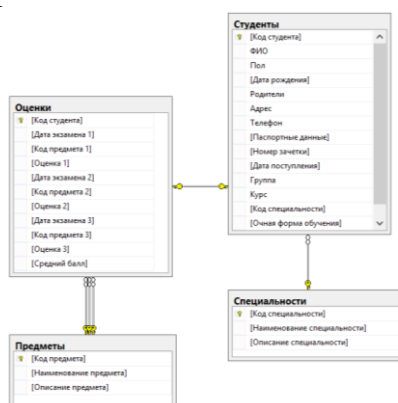
```

Сообщения

Выполнение команд успешно завершено.

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

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



8. 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.

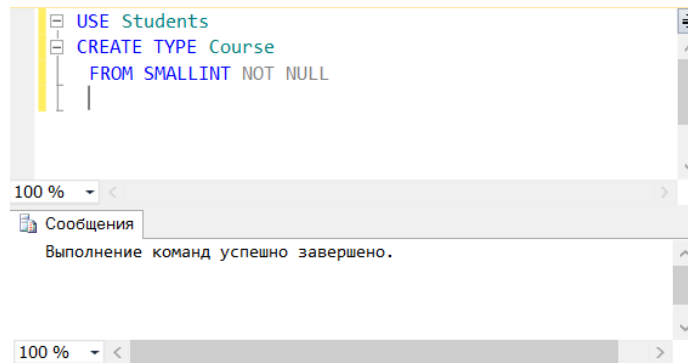
```

CREATE TABLE Студенты
--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,
    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
)

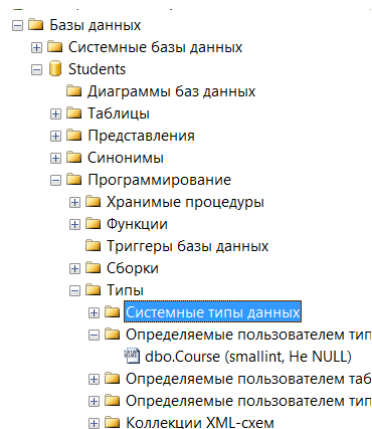
```


9. Создание псевдонимного типа данных

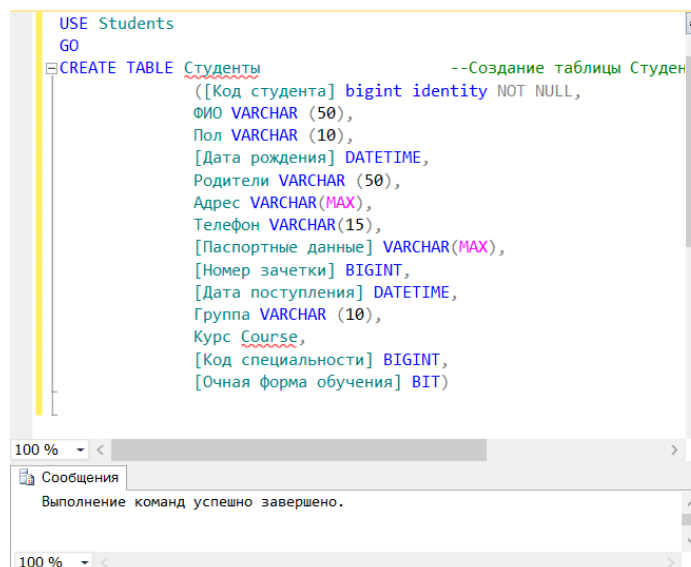
В следующем примере показано использование инструкции CREATE TYPE для создания псевдонимного типа данных Course на основе стандартного типа данных SMALLINT.



Все псевдонимные типы данных сохраняются в папке «Типы» в узле соответствующей базы данных в обозревателе объектов сервера.



После создания псевдонимного типа данных его можно присвоить столбцу таблицы. В следующем примере показано создание таблицы Студенты, в которой столбцу Курс присвоен тип данных Course.



2. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Назовите виды объектов базы данных. Приведите примеры.
2. Назовите инструкции создания базы данных и создания таблицы.
3. Дайте определение понятию «владелец базы данных».
4. Назовите системные процедуры, которые используются для отсоединения базы данных от сервера и присоединения.
5. Поясните смысл предложения DEFAULT и параметра IDENTITY, которые указываются в спецификации столбца при создании таблицы.
6. Дайте определение понятию «ограничения для обеспечения целостности (integrity constraints)».
7. Назовите типы ограничений для обеспечения целостности, которые предоставляются СУБД.
8. Перечислите инструкции языка DDL, с помощью которых определяются декларативные ограничения целостности.
9. Поясните разницу между декларативными ограничениями уровня столбцов и уровня таблицы.
10. Назовите опцию, посредством которой может быть явно присвоено имя декларативному ограничению.
11. Назовите категории декларативных ограничений.
12. Назовите основные отличия на столбец UNIQUE и столбец PRIMARY KEY.
13. Назовите следующие предложения в инструкции CREATE TABLE или ALTER TABLE: создание потенциального ключа, создание первичного ключа, создание проверочного ограничения, создание внешнего ключа.
14. Поясните понятие ссылочной целостности.

ДОМАШНЕЕ ЗАДАНИЕ

[1], страницы 111-145

ЛИТЕРАТУРА

1. Петкович, Д. Microsoft SQL Server 2012. Руководство для начинающих: пер. с английского / Д. Петкович. – СПб.: БХВ-Петербург, 2013. – 816 с.: ил.
2. Сеть разработчиков Microsoft [Электронный ресурс]. – Режим доступа: <https://msdn.microsoft.com/ru-ru/library>

Преподаватель

С.В. Банцевич

Рассмотрено на заседании цикловой
комиссии программного обеспечения
информационных технологий №10
Протокол № от « » _____ 2017
Председатель ЦК С.В. Банцевич