Тема: Создание базы данных и таблиц БД

- Каждая БД как минимум состоит из двух файлов, один для данных, другой для журнала транзакций.

Create Database <database name>

Тип данных	Краткое описание
bigint	Целочисленный тип данных, занимающий 8 байт
float	Нецелочисленный тип данных приблизительной точности
ntext	Текстовые данные Unicode длиной до 1 Гбайта
int	Целочисленный тип данных, занимающий 4 байта
real	Нецелочисленный тип данных приблизительной точности
binary	Двоичные данные фиксированной длины до 8000 байт
smallint	Целочисленный тип данных, занимающий 2 байта
datetime	Дата и время высокой точности (8-байтовый)
varbinary	Двоичные данные переменной длины до 8000 байт
tinyint	Целочисленный тип данных, занимающий 1 байт
smalldatetime	Дата и время низкой точности (4-байтовый)
image	Двоичные данные длиной до 2 Гбайт
bit	Один бит, принимает значение либо 0, либо 1
char	Символьные данные не Unicode фиксированной длины до 8000 символов
decimal	Нецелочисленный тип данных фиксированной точности

Тип данных	Краткое описание
varchar	Символьные данные не Unicode переменной длины до 8000 символов
timestamp	Временной штамп или версия строки
numeric	Нецелочисленный тип данных фиксированной точности
text	Текстовые данные не Unicode длиной до 2 Гбайт
sql_variant	Тип данных, позволяющий хранить значения других типов данных
money	Денежный тип данных высокой точности (8-байтовый)
nchar	Символьные данные Unicode фиксированной длины до 4000 символов
smallmoney	Денежный тип данных низкой точности (4-байтовый)
nvarchar	Символьные данные Unicode переменной длины до 4000 символов
uniqueidentifier	Тип данных, предназначенный для хранения глобальных уникальных идентификаторов

Создание таблицы:

CREATE TABLE <имя_таблицы>
(<определение_столбца>,
<определение_столбца>,
<определение_столбца>)

В скобках указываются все предложения данной инструкции, определяющие отдельные элементы таблицы или ограничения целостности.

<определение_столбца>::=

<имя_столбца><тип_столбца>[<ограничение_столбца>]

<u>Ограничения</u>-
Null
первичный ключ
внешний ключ
уникальный столбец
проверочной ограничение Chek
значение по умолчанию

с помощью параметра ***CONSTRAINT*** создается поименованное ограничение, так как модификация или удаление любого ограничения происходит по его имени.

Таблица может содержать ограничение **PRIMARY KEY** только для одного столбца или для группы столбцов.

UNIQUE может присутствовать любое количество раз. Null-значения допустимы.

<Ограничение_столбца> состоит из:

[CONSTRAINT<имя_ограничения>]-ключевое слово, после которого указывается название ограничения.

{[DEFAULT<выражение>]- значение по умолчанию для столбца.

|[NULL]|[NOT NULL]- разрешающие или запрещающие null-значения

|[PRIMARY KEY|UNIQUE]- первичный ключ, уникальность значения.

| [FOREIGN KEY…REFERENCES <имя_таблицы>]
[(<имя_столбца>[,…,n])]- описывает внешний ключ.

|[ON DELETE {CASKADE|NO ACTION}]-действия при удалении: 1) удаляют и из зависимой таблицы; 2) игнорируют удаление и сообщение об ошибке.

|[ON UPDATE {CASKADE|NO ACTION}]- действия при изменении аналогичные действиям при удалении.

|[CHECK()]}-ограничение целостности, инициирующее контроль вводимых в столбец значений.

NO ACTION устанавливается по умолчанию.

Создать таблицу Студент с первичным ключом Код студента и данными: ФИО и группа.

```
CREATE TABLE Студент (код_студента
    int PRIMARY KEY, фамилия CHAR
    (15), имя CHAR (10), отчество CHAR
    (15), группа CHAR (4))
```

Создать таблицу Предмет с первичным ключом Код предмета и название предмета, которое будет уникальным значением атрибута Преподаватель должно быть определено для каждого предмета.

```
CREATE TABLE предмет (код_предмета
    INT CONSTRAINT PK_код_предмета
    PRIMARY KEY (код_предмета),
    название_предмета CHAR (20)
    UNIQUE, преподаватель CHAR (30)
    NOT NULL)
```

В следующем примере производится создание новой таблицы со свойством IDENTITY для получения автоматически увеличивающегося идентификационного номера.

```
CREATE TABLE  new_employees
 ( id_num int IDENTITY(1,1) PRIMARY KEY , fname varchar (20), lname varchar(30) );
```

Задание

- 1 Создать БД со своей фамилией
- 2 Создать таблицу Студент, указав PRIMARY KEY сразу после ключевого атрибута (вначале)
- 3 Создать таблицу Студент1, указав PRIMARY KEY (в конце) после всех столбцов
- 4 Создать таблицу Студент2, указав ограничение CONSTRAINT
- Для PRIMARY KEY сразу после ключевого атрибута (вначале)

- *Создать таблицу Оценки с полями Код_предмета, Код_студента, оценка, дата_сдачи. Они должны быть определены, при чем Код_студента и Код_предмета были внешними ключами, а оценку можно вводить от 2 до 5.*

- CREATE TABLE Оценки (код_предмета INT CONSTRAINT FK_KP FOREIGN KEY (код_предмета) REFERENCES предмет (код_предмета), код_студента INT CONSTRAINT FK_KC FOREIGN KEY (код_студента) REFERENCES студент (код_студента), оценка INT (1) CHECK ((оценка>=2 AND оценка<=5) дата_сдачи DATE)