# Создание таблиц и обеспечение целостности данных

## Цели урока

В этом уроке вы узнаете как:

- создавать таблицы
- вставлять строки в таблицы
- удалять таблицы
- создавать типы данных, определяемые пользователем
- перечислять различные типы целостности данных
- добавлять следующие ограничения в таблицы:
  - PRIMARY KEY
  - UNIQUE
  - FOREIGN KEY
  - CHECK
  - DEFAULT

# Создание таблиц и обеспечение целостности данных

## CPC 5_2

1. Что такое подзапрос?

2. Составить команду SQL, которая выводит список кадровых агентов, проживающих в том же штате, что и внешний кандидат Barbara Johnson.

3. Указать, истинно выражение или ложно.

   Список столбца команды SELECT подзапроса, введенного с оператором сравнения, может включать только один столбец.

1. Что такое связанный подзапрос?

2. Сколько максимально возможных столбцов в таблице?

# Создание таблиц и обеспечение целостности данных

## Создание таблицы

***Формулировка задачи:***

***Требуется сохранить подробности о газете, публикующей объявления для Tebisco, Inc., в базу Recruitment (прием на работу).***

| Атрибуты | Данные |
|---|---|
| Newspaper Code | 0001 |
| Newspaper Name | Texas Times |
| Region (область) | Texas (Техас) |
| Type of Newspaper (тип газеты) | General (общая) |
| Contact Person (контактное лицо) | Jackson Demello |
| HO Address (адрес главного офиса) | 4723 West Alabama |
| City (город) | Houston (Хьюстон) |
| State (штат) | Texas (Техас) |
| Zip (почтовый индекс) | 77015-4568 |
| Country Code (код страны) | 001 |
| Fax (факс) | (713)451-6797 |
| Phone (телефон) | (713)451-6850 |

***Код газеты и ее название не могут быть не указаны.***

# Создание таблиц и обеспечение целостности данных

## Решение

1. *Составить команду создания таблицы.*
2. *Создать таблицу в базе.*
3. *Проверить, создана ли эта таблица.*
4. *Вставить строку в таблицу.*

# Создание таблиц и обеспечение целостности данных

## Составить команду создания таблицы

- *Таблицы: Таблица – это объект базы данных для хранения данных*
  - *Данные в таблице организованы в строки и столбцы*
  - *Каждая строка в таблице представляет уникальную запись, а каждый столбец – атрибут этой записи*
  - *Команда CREATE TABLE применяется для создания таблицы*
  - *Синтаксис*

```
CREATE TABLE имя_таблицы
(имя_столбца типДанных [NULL | NOT NULL]
[IDENTITY (SEED,INCREMENT)],
имя_столбца типДанных …)
[ON {группаФайлов} | DEFAULT]
```

*Результат*

- *Newspaper – имя создаваемой таблицы*

# Создание таблиц и обеспечение целостности данных

## Создать таблицу в базе

*Действие*

- *В окне Query Analyzer (Анализатор запроса) ввести данную команду CREATE TABLE*

```
CREATE TABLE NewsPaper
(cNewsPaperCode char(4) NOT NULL,
cNewsPaperName char(20) NOT NULL,
vRegion varchar(20),
vTypeOfNewsPaper varchar(20),
vContactPerson varchar(35),
vHOAddress varchar(35),
cCity char(20),
cState char(20),
cZip char(10),
cCountryCode char(3),
cFax char(15),
cPhone char(15))
```

- *Нажать F5 для выполнения команды*

# Создание таблиц и обеспечение целостности данных

## Проверить, создана ли эта таблица

- *Для просмотра структуры таблицы можно использовать команду sp_help*
- *Синтаксис*

```
sp_help имя_таблицы
```

*Действие*

- *В окне Query Analyzer набрать:*

```
sp_help Newspaper
```

- *Нажать F5 для выполнения*

# Вставить строку в таблицу

***Команда INSERT***

- ***После того, как создана структура таблицы, в нее можно вставлять данные. Для этого применяется команда INSERT***
- ***Синтаксис***

```
INSERT [INTO] имя_таблицы [список_столбцов]
VALUES (список_значений)
```

# Вставить строку в таблицу (продолжение)

*Действие*

- *В окне Query Analyzer (Анализатор запроса) ввести команду INSERT:*

```
INSERT Newspaper
VALUES('0001', 'Texas Times', 'Texas',
'General', 'Jackson Demello',
'4723 West Alabama', 'Houston', 'Texas',
'77015-4568', '001', '(713)451-6797',                '(713)451-6850')
```

- *Нажать F5 для выполнения*

# Практика-Создание таблицы

***Формулировка задачи***

***Требуется сохранить подробности о университетах, к которым обратилась Tebisco, Inc. для приема на работу. Пример строки:***

| Атрибут | Данные |
|---|---|
| **College Code** *(код университета)* | **0002** |
| **College Name** *(Название университета)* | **Cromwell College** |
| **College Address** *(Адрес университета)* | **4010 Gartner Ave** |
| **City** *(город)* | **Abilene** |
| **State** *(штат)* | **Texas (Texac)** |
| **Zip** *(почтовый индекс)* | **79605-4123** |
| **Phone** *(телефон)* | **(915)692-6628** |

***Код университета и его название не могут быть не указаны.***

# Демонстрация-Удаление таблицы

***Формулировка задачи***

***Удалить таблицу Newspaper из базы.***

# Создание таблиц и обеспечение целостности данных

## Решение

1. *Составить оператор удаления таблицы из базы.*
2. *Удалить таблицу Newspaper из базы.*
3. *Проверить, что таблица удалена из базы.*

# Создание таблиц и обеспечение целостности данных

Составить команду удаления таблицы из базы.

*Команда DROP TABLE*

- *удаляет таблицу Newspaper из базы*
- *Синтаксис*

```
DROP TABLE имя_таблицы
```

*Результат*

- *Команда удаления таблицы Newspaper :*

```
DROP TABLE Newspaper
```

# Создание таблиц и обеспечение целостности данных

## Удалить таблицу Newspaper из базы

***Действие***

- ***В окне Query Analyzer набрать:***
  ***DROP TABLE Newspaper***
- ***Нажать F5 для выполнения***

# Создание таблиц и обеспечение целостности данных

## Проверить, что таблица удалена из базы

- *Для просмотра структуры таблицы набрать следующую команду:*

```
sp_help Newspaper
```

*Действие*

- *Для просмотра структуры таблицы набрать следующую команду:*
- *В окне Query Analyzer набрать:*

```
sp_help Newspaper
```

- *Нажать F5 для выполнения*

# Типы данных, определенные пользователем

- Создание типа данных, определенного пользователем
  - Тип данных, определенный пользователем, создается процедурой sp_addtype , записанной в системе
  - Синтаксис

```
sp_addtype имя, [системный_тип_данных]
[,'тип_NULL']
```

- Игнорирование типа данных, определенного пользователем
  - Тип данных, определенный пользователем, игнорируется процедурой sp_droptype, записанной в системе
  - Синтаксис

```
sp_droptype type
```

# Демонстрация-Создание типов данных, определенных пользователем

***Формулировка задачи:***

***Таблицы NewsAd и Newspaper - неодинакового типа для атрибута cNewspaperCode. Создать тип данных typNewspaperCode, определенный пользователем, который может применяться для создания таблиц NewsAd и Newspaper. Создать табл. NewsAd, в которой код газеты типа typNewspaperCode.***

# Создание таблиц и обеспечение целостности данных

## Решение

1. *Определить несовместимость структур таблиц*
2. *Определить тип данных, определенный системой, который может преобразовываться в тип, определенный пользователем*
3. *Определить имя типа, определенного пользователем*
4. *Создать тип данных, определенный пользователем*
5. *Проверить, создан ли этот тип данных*
6. *Создать таблицу NewsAd с новым типом данных*

# Создание таблиц и обеспечение целостности данных

## Определить несовместимость структур таблиц

***Типы данных, определенные пользователем***

- ***могут применяться для устранения несовместимости структур таблиц при использовании разных системных типов двумя атрибутами, которые должны иметь одинаковый системный тип.***

***Результат***

- ***Атрибут cNewspaperCode в табл. Newspaper (Газета) является типом char(4). Атрибут cNewspaperCode в табл. NewsAd является типом varchar(2). Тип и длина обоих атрибутов различны. Это увеличивает несовместимость структуры таблицы.***

# Создание таблиц и обеспечение целостности данных

Определить тип данных, определенный системой, который может переводиться в тип данных, определенный пользователем

***Результат***

- ***Желательно, чтобы тип обоих атрибутов, определенный системой, был char(4).***

# Создание таблиц и обеспечение целостности данных

## Определить имя типа, определенного пользователем

- ***Имя типа может иметь приставку 'typ' для определения типа, определенного пользователем***

***Результат***

- ***Имя типа, определенного пользователем, - typNewspaperCode***

# Создание таблиц и обеспечение целостности данных

## Создать тип данных, определенный пользователем

*Действие*

- *В окне Query Analyzer набрать:*

```
sp_addtype typNewspaperCode, 'char(4)'
```

- *Выполнить запрос, нажав 'Execute Query'*

# Создание таблиц и обеспечение целостности данных

## Проверить, что таблица создана

- *Процедура sp_help, записанная в системе, дает особую информацию о заданном объекте*
- *Синтаксис*

```
sp_help datatype_name
```

*Действие*

- *В окне Query Analyzer набрать:*

```
sp_help typNewspaperCode
```

- *Нажать F5 для выполнения*

# Создание таблиц и обеспечение целостности данных

## Создать таблицу NewsAd с новым типом данных

*Действие*

- *В окне Query Analyzer набрать:*

```
CREATE TABLE NewsAd
(
cNewsAdNo char(4) NOT NULL,
cNewspaperCode typNewspaperCode NOT NULL,
dAdStartDate datetime,dDeadline datetime
)
```

- *Нажать F5 для выполнения*

# Целостность данных

- Целостность данных обеспечивает совместимость и точность данных, хранящихся в базе. В широком смысле она делится на следующие четыре категории:
  - Целостность сущностей-объектов
  - Целостность домена
  - Целостность ссылочных данных
  - Целостность, определяемая пользователем
- Целостность сущностей-объектов
  - гарантирует то, что каждая строка может однозначно определяться по атрибуту "первичный ключ"
- Целостность домена
  - гарантирует, что хранить в столбце допускается только действительный диапазон значений

# Целостность данных (продолжение)

- Целостность ссылочных данных
  - гарантирует, что значения внешнего ключа соответствуют значению первичного ключа
- Пользовательская целостность
  - относится к набору правил, задаваемых пользователем и не принадлежащих к категориям Целостности сущностей-объектов, домена и ссылочных данных

# Создание ограничений

- Ограничения создаются для обеспечения целостности данных
- Ограничения определяют правила для поддержания соответствия и верности данных
- Ограничения создаются при создании таблицы или добавляются позже
- Ограничения могут вводиться на двух уровнях:
  - уровень столбца
  - уровень таблицы

# Создание ограничений (продолжение)

- Ограничения создаются для обеспечения целостности данных
- Ограничение может создаваться одной из следующих команд:
- CREATE TABLE

  CREATE TABLE имя_таблицы
  имя_столбца CONSTRAINT имя_ограничения тип_ограничения
[,CONSTRAINT имя_ограничения тип_ограничения]
- команда ALTER TABLE

ALTER TABLE имя_таблицы
[WITH CHECK | WITH NOCHECK]
ADD CONSTRAINT имя_ограничения имя_ограничения

# Создание ограничений (продолжение)

- Сброс ограничений
  - Ограничение можно сбросить командой ALTER TABLE в Query Analyzer (Анализаторе запроса)
  - Синтаксис

```
ALTER TABLE имя_таблицы
DROP CONSTRAINT имя_переменной
```

# Создание ограничений (продолжение)

- Типы ограничений
  - Ограничение PRIMARY KEY
  - определяется по одному или нескольким столбцам, значения которых однозначно опознают строки в таблице
  - Обеспечивает целостность сущностей-объектов
  - Синтаксис

```
[CONSTRAINT имя_ограничения PRIMARY KEY
[CLUSTERED|NONCLUSTERED](имя_столбца [,
имя_столбца [,имя_столбца [, …]]])]
```

# Создание ограничений (продолжение)

- Ограничение UNIQUE
  - накладывает единственность на столбцы непервичного ключа
  - Можно создать несколько ограничений UNIQUE на таблицу
  - Синтаксис

```
[CONSTRAINT имя_ограничения UNIQUE
[CLUSTERED | NONCLUSTERED](имя_столбца [,                имя_столбца
[, имя_столбца [, …]]])
```

# Создание ограничений (продолжение)

- Ограничение FOREIGN KEY
  - удаляет несоответствие в двух таблицах, если данные в одной таблице зависят от данных в другой
  - Синтаксис

```
[CONSTRAINT имя_ограничения FOREIGN KEY
(имя_столбца [, имя_столбца [, …]])
        REFERENCES имя_таблицы (имя_столбца [,
        имя_столбца [, …]])]
```

# Создание ограничений (продолжение)

- Ограничение CHECK
  - накладывает целостность домена, ограничивая значения, вставляемые в столбец
  - Синтаксис

```
[CONSTRAINT имя_ограничения] CHECK (выражение)
```

- Ограничение DEFAULT
  - Применяется для присвоения постоянного значения столбцу
  - Для столбца может создаваться только одно ограничение DEFAULT
  - Этот столбец не может быть столбцом IDENTITY
  - Синтаксис

```
[CONSTRAINT имя_ограничения] DEFAULT
(выражение_константа | NULL)
```

# Демонстрация - Применение ограничений

***Формулировка задачи***

***Таблицы Newspaper и NewsAd завершены.***

***Создать таблицу Newspaper по следующим правилам целостности данных:***

- ***Атрибут cNewspaperCode должен быть первичным ключом***
- ***Атрибут cPhone (телефон) должен быть формата ([0-9][0-9][0-9])[0-9][0-9][0-9]-[0-9][0-9][0-9][0-9])***
- ***Атрибут cCountrycode (код страны) должен быть '001' по умолчанию***

***Изменить таблицу NewsAd следующим образом:***

- ***cNewsAdNo (№ объявления в газете) должен быть первичным ключом***
- ***cNewspaperCode (код газеты) должен быть внешним ключом***

# Создание таблиц и обеспечение целостности данных

## Решение

1. *Определить способ наложения целостности данных.*
2. *Составить команду создания таблицы.*
3. *Создать таблицу с ограничениями.*
4. *Проверить ограничения, вставляя данные.*

# Создание таблиц и обеспечение целостности данных

## Определить способ наложения целостности данных

*Результат*

- *Целостность данных налагается ограничениями*
- *Для таблицы Newspaper (газета):*
  - *форму номера телефона можно задать ограничением CHECK*
  - *код страны можно задать ограничением DEFAULT*
  - *код газеты можно задать первичным ключом, применяя ограничение PRIMARY KEY*
- *Для таблицы NewsAd:*
  - *столбец cNewsAdNo можно задать первичным ключом, применяя ограничение PRIMARY KEY*
  - *атрибут cNewspaperCode можно задать внешним ключом, применяя ограничение FOREIGN KEY*

# Создание таблиц и обеспечение целостности данных

## Составить команду создания таблицы

***Результат***

- ***Желательно, чтобы команда создания таблицы Newspaper была следующей:***

```
CREATE TABLE Newspaper
(cNewspaperCode typNewspaperCode CONSTRAINT
pkNewspaperCode PRIMARY KEY,
cNewspaperName char(20) NOT NULL,
vRegion varchar(20),
vTypeOfNewspaper varchar(20),
vContactPerson varchar(35),
vHOAddress varchar(35),
cCity char(20),
cState char(20),
cZip char(10),
cCountryCode char(3) CONSTRAINT defCountryCode
DEFAULT('001'),
cFax char(15),
```

# Создание таблиц и обеспечение целостности данных

## Составить команду создания аблицы (продолжение)

```
cPhone char(15) CONSTRAINT chkPhone
CHECK(cPhone
LIKE('([0-9][0-9][0-9])[0-9][0-9][0-9]-[0-9][0-
9][0-9][0-9]')))
```

- *Желательно, чтобы команды изменения таблицы NewsAd были следующими:*

```
ALTER TABLE NewsAd
    ADD CONSTRAINT pkNewsAdNo PRIMARY KEY
    (cNewsAdNo)
    ALTER TABLE NewsAd
    ADD CONSTRAINT fkNewspaperCode
    FOREIGN KEY (cNewspaperCode)
    REFERENCES Newspaper(cNewspaperCode)
```

# Создание таблиц и обеспечение целостности данных

## Создать таблицу с ограничениями

*Действие*

- *В окне Query Analyzer набрать запрос*

```
CREATE TABLE Newspaper
(
cNewspaperCode typNewspaperCode CONSTRAINT
pkNewspaperCode PRIMARY KEY,
cNewspaperName char(20) NOT NULL,
vRegion varchar(20),
vTypeOfNewspaper varchar(20),
vContactPerson varchar(35),
vHOAddress varchar(35),
cCity char(20),
cState  char(20),
cZip char(10),
```

# Создание таблиц и обеспечение целостности данных

## Создать таблицу с ограничениями (продолжение)

*Действие*

```
cCountryCode char(3) CONSTRAINT defCountryCode
    DEFAULT('001'),
    cFax char(15),
    cPhone char(15) CONSTRAINT chkPhone CHECK(cPhone
    LIKE('([0-9][0-9][0-9])[0-9][0-9][0-9]-[0-9][0-9][0-9][0
    -9]'))
    )
ALTER TABLE NewsAd
ADD CONSTRAINT pkNewsAdNo PRIMARY KEY (cNewsAdNo)
ALTER TABLE NewsAd
ADD CONSTRAINT fkNewspaperCode FOREIGN KEY (cNewspaperCode)
REFERENCES Newspaper(cNewspaperCode)
```

- *Выполнить команды, кликнув на Execute Query*

# Проверить ограничения, вставляя данные

***Проверить ограничение, вставляя данные в табл. Newspaper***

| № теста | Атрибут | Действие: Значение в команде INSERT | Результат |
|---|---|---|---|
| 1 | **cPhone** | 3445AB323 | Строка не вставлена, т.к. телефон имеет символьные данные |
| 2 | **cPhone** | (212)345-2467 | Строка вставлена, т.к. форма телефона верна |
| 4 | **cCountrycode** | 005 | Строка вставлена с 005 в атрибуте **cCountryCode** |
| 5 | **cNewsPaperCode** | 0001 (уже есть в табл.**Newspaper**) | При вставке строки выводится сообщение об ошибке, т.к. 0001 уже есть для **cNewspaperCode** в табл.**Newspaper** |
| 6 | **cNewsPaperCode** | 0090 (нет в табл.**Newspaper**) | Строка вставлена, т.к. 0090 для **cNewspaperCode** нет в табл.**Newspaper** |

# Создание таблиц и обеспечение целостности данных

## Проверить ограничения, вставляя данные (продолжение)

***Проверить ограничение, вставляя данные в табл. NewsAd***

| Контрольный пример | Атрибут | Действие: Значение в команде SELECT | Результат |
|---|---|---|---|
| 1 | **cNewsAdNo** | 0001 (уже есть в табл.**NewsAd**) | При вставке строки выводится сообщение об ошибке, т.к. 0001 уже есть для **cNewsAdNo** в табл.**NewsAd** |
| 2 | **cNewsAdNo** | 0035 (нет в табл.**NewsAd**) | Строка вставлена, т.к. 0035 нет в табл.**NewsAd** |
| 3 | **cNewsPaperCode** | 0045 (нет в табл.**Newspaper**) | При вставке строки выводится сообщение об ошибке, т.к. 0045 нет для **cNewspaperCode** в табл.**Newspaper** |
| 4 | **cNewsPaperCode** | 0001 (есть в табл.**Newspaper**) | Строка вставлена, т.к. 0001 есть для **cNewpaperCode** в табл.**NewsAd** |

# Демонстрация - Применение ограничений

***Формулировка задачи***

***Создать таблицу College по следующим правилам целостности данных:***

- ***cCollegeCode (код университета) должен быть первичным ключом***
- ***телефон должен быть формата ([0-9][0-9][0-9])[0-9][0-9][0-9]-[0-9][0-9][0-9][0-9])***
- ***cCity (город) должен быть 'New Orleans' по умолчанию***

***Создать таблицу CampusRecruitment так, чтобы она отвечала следующим правилам целостности данных:***

- ***столбец cCampusRecruitmentCode должен быть первичным ключом***
- ***столбец cCollegeCode должен быть внешним ключом***

# Создание таблиц и обеспечение целостности данных

## Выводы

Из этого урока Вы узнали, что:

- Таблица – это объект базы данных для хранения данных
- Таблица может быть создана командой CREATE TABLE
- Команда INSERT применяется для вставки данных в таблицу
- Команда DROP TABLE применяется для удаления таблицы
- Тип данных, определенный пользователем, создается пользователем и основан на типе данных системы
- Тип данных, определенный пользователем, создается процедурой sp_addtype , записанной в системе
- Тип данных, определенный пользователем, сбрасывается процедурой sp_droptype, записанной в системе
- sp_help дает информацию об объекте базы данных или типе данных, определяемом пользователем
- Целостность данных обеспечивает полность, точность и надежность данных, хранящихся в базе

# Выводы (продолжение)

- Целостность данных имеет категории: целостность сущностей-объектов, целостность домена, целостность ссылочных данных и целостность, определяемая пользователем
- Целостность данных можно наложить ограничениями
- Ограничения – это правила, задаваемые на уровне таблицы или столбца
- Ограничение может быть создано командами CREATE TABLE или ALTER TABLE
- Ограничение можно сбросить командой ALTER TABLE или сбрасывая таблицу
- Ограничения имеют классы PRIMARY, FOREIGN, UNIQUE, CHECK и DEFAULT