**Лабораторная работа № 15. Создание таблиц средствами SQL**

**2 часа**

**Цель:** получение навыков создания таблиц средствами T-SQL.

**15.1 Теоретические положения**

Процесс создания таблицы начинается с проектирования ее будущей структуры. В процессе проектирования необходимо решить следующие вопросы:

1) для хранения каких данных предназначена создаваемая таблица?

2) каким образом будет обеспечиваться целостность данных в ней. Для этого следует определить ограничения на значения колонок (constraints).

SQL Server позволяет управлять значениями колонок при помощи следующих механизмов.

1. Определение первичного ключа (Primary Key).

2. Определение внешнего ключа (Foreign Key).

3. Создание уникальных колонок (Unique) в нескольких колонках таблицы, помимо первичного ключа.

4. Наложение проверочных ограничений на значения колонок (Check).

5. Определение значений по умолчанию (Defaults).

6. Какие колонки могут содержать неопределенные значения (NULL).

Для создания таблицы используется следующая инструкция T-SQL.

CREATE TABLE [ database [owner]. | owner. ] name

( { <column\_definition>

| column\_name AS computed\_col\_expr

|<table\_constraint>} )

[ON {filegroup | DEFAULT}]

Ключевое слово ON позволяет указать файловую группу, в которой будет располагаться таблица. Здесь есть две возможности: либо явно указать имя файловой группы (причем она уже должна существовать в базе данных), либо использовать ключевое слово DEFAULT, которое предписывает системе расположить таблицу в файловой группе по умолчанию.

Определение каждой колонки таблицы, в синтаксисе команды обозначенное как <column\_definition>, имеет следующий формат:

{column\_name data\_type}

[DEFAULT constant

| [IDENTITY [(seed, increment) [NOT FOR REPLICATION]]]]

[ROWGUIDCOL]

[<column\_constraint>]

Подобным образом необходимо описать каждую колонку в таблице. Прежде всего, следует определить имя колонки (column\_name), а также тип хранимых в ней данных (data\_type). При описании могут быть использованы следующие ключевые слова.

DEFAULT – определяет значение по умолчанию (constant\_expression), используемое, если при вводе строки явно не указано другое значение.

IDENTITY – предписывает системе осуществлять заполнение колонки автоматически. При этом также указать начальное значение (seed) и приращение (increment). В случае, когда указано NOT FOR REPLICATION, эта колонка не будет автоматически заполняться для строк, вставляемых в таблицу в процессе репликации, так что эти строки сохранят свои значения.

ROWGUIDCOL – данная колонка будет использоваться для хранения глобального идентификационного номера.

Кроме того, для колонки можно определить ограничения на значения. Это делается следующим образом:

[CONSTRAINT constraint\_name]

{ [ NULL | NOT NULL] | [{ PRIMARY KEY | UNIQUE }

[CLUSTERED | NONCLUSTERED]

[WITH FILLFACTOR = fillfactor]

[ON { filegroup | DEFAULT}

| [ [FOREIGN KEY]

REFERENCES ref\_table ]

Наложение ограничения на значение колонки должно начинаться с ключевого слова CONSTRAINT, после которого необходимо указать имя ограничения на зна­чение. Ограничения могут носить произвольные названия, но, как правило, для применяются следующие префиксы:

* "PK\_" – для PRIMARY KEY
* "FK\_" – для FOREIGN KEY
* "CK\_" – для CHECK
* "UQ\_" – для UNIQUE
* "DF\_" – для DEFAULT

В принципе необязательно задавать имена ограничений, при установке соответствующих атрибутов SQL Server автоматически определяет их имена. Но, зная имя ограничения, мы можем к нему обращаться, например, для его удаления.

Для каждого ограничения желательно указать, допустимо ли для колонки значение NULL, выбрав соответствующее ключевое слово (NULL или NOT NULL). После этого требуется определить тип ограничения.

PRIMARY KEY – определяет колонку как первичный ключ таблицы. В качестве альтернативы можно определить колонку как уникальную, воспользовавшись ключевым словом UNIQUE. При необходимости можно также указать, будет ли индекс, создаваемый для данного ограничения, кластерным (ключевое слово CLUSTERED) или некластерным (NONCLUSTERED). Однако кластерный индекс можно определить только для одного ограничения, поэтому требуется решить, с каким ограничением (первичный ключ или уникальная колонка) он будет использоваться. Если создается индекс, необходимо также указать степень заполнения его страниц (ключевое слово WITH FILLFACTOR).

FOREIGN KEY – определяет колонку как внешний ключ таблицы. Одновременно, используя ключевое слово REFERENCES, необходимо указать имя таблицы, с которой будет связана создаваемая таблица. Дополнительно потребуется указать ее колонки, которые будут связаны с данной колонкой.

Пример создания таблицы с ограничениями на значения столбцов:

CREATE TABLE Customers

(

Id INT CONSTRAINT PK\_Customer\_Id PRIMARY KEY IDENTITY,

Age INT

CONSTRAINT DF\_Customer\_Age DEFAULT 18

CONSTRAINT CK\_Customer\_AgeCHECK(Age >0 AND Age < 100),

)

Следует заметить, что можно определить виртуальную колонку, которая будет производной от других колонок. При этом на хранение ее значений не требуется физической памяти, поскольку они в любой момент могут быть получены для любой строки путем вычислений. Определение виртуальной колонки имеет следующий формат:

имя\_колонки AS выражение

Выражение может включать имена колонок, функции и арифметические операции.

Более подробно вопросы создания таблиц описаны в конспекте лекций и в [2, 3, 4, 6, 15].

**Задание**

Необходимо создать средствами Т-SQL три таблицы, обеспечивающие возможность сохранения копий строк из других таблиц при удалении данных. Реализовать три триггера, обеспечивающие запись данных в созданные резервные таблицы при удалении из основных таблиц базы данных.

**Содержание отчета:** тема и цель работы; прокомментированный SQL-код выполнения задания.

**Контрольные вопросы**

1 Перечислить этапы проектирования структуры таблиц.

2 С помощью каких команд T-SQL можно создать, изменить или удалить таблицу?

3 Для чего нужны уникальные колонки, проверочные ограничения на значения колонок, идентификационные колонки?

4 Охарактеризовать механизмы управления значениями колонок (Primary Key, Foreign Key, Unique, Check, Defaults, NULL).