**Тема: Язык SQL. Создание таблиц.**

**Источники:** 1. Основы T-SQL. DDL. URL: https://metanit.com/sql/sqlserver/3.1.php.

2. Основы T-SQL.DML. URL: https://metanit.com/sql/sqlserver/4.1.php.

Язык SQL – это язык структурированных запросов, который позволяет формировать весьма сложные запросы к базе данных.

**Определение:** Запрос – это вопрос к базе данных, возвращающий запись или множество записей, удовлетворяющих вопросу.

В зависимости от характера выполняемых действий операторы SQL можно разделить на следующие группы:

**1. Операторы определения данных:**

* + - Create Table – создание новой таблицы базы данных;
    - DropTable – удаление таблицы из базы данных;
    - AlterTable – изменение структуры существующей таблицы или ограничение целостности, задаваемой для данной таблицы;
    - CreateView – создание виртуальной таблицы, соответствующей некоторому SQL-запросу;
    - DropView – удаление ранее созданного представления;
    - CreateIndex – создание индекса для некоторой таблицы для обеспечения быстрого доступа по атрибутам, входящим в индекс;
    - DropIndex – удаление ранее созданного индекса.

**2. Операторы манипулирования данными:**

* Delete – удаление одной или нескольких строк из базовой таблицы, соответствующих условиям фильтрации;
  + - Insert – вставка одной строки в базовую таблицу;
    - Update – обновление значения одного или нескольких столбцов в одной или нескольких строках, соответствующих условиям фильтрации.

**3. Оператор запросов:** Select – оператор выбора одной или нескольких строк из одной или нескольких таблиц, удовлетворяющих условиям фильтрации.

**4. Операторы управления транзакциями:**

* + - Commit – завершение комплексной, взаимосвязанной обработки информации, объединенной в транзакцию;
    - RollBack – отменение изменений, проведенных в ходе выполнения транзакции;
    - SavePoint – сохранение промежуточного состояния базы данных, помечает его для того, чтобы можно было в дальнейшем к нему вернуться.

**5. Операторы администрирования данными:**

* + - AlterDatabase – изменение набора основных объектов в базе данных;
    - AlterDbarea – изменение ранее созданной области хранения;
    - AlterPassword – изменение пароля для всей базы данных;
    - CreateDatabase – создание новой базы данных;
    - CreateDbarea – создание новой области хранения базы данных;
    - DropDatabase – удаление базы данных;
    - DropDbarea – удаление области хранения базы данных;
    - Grant – предоставление права доступа к базе данных или отдельным ее элементам;
    - Revoke – лишение права доступа к базе данных или отдельным ее элементам.

**6. Операторы управления курсором:**

* + - Declare – определение курсора для запроса;
    - Open – открытие курсора;
    - Feth – установка курсора на определенную запись и считывание ее;
    - Close – закрытие курсора;
    - Prepare – генерация плана выполнения запроса;
    - Execute – выполнение ранее сгенерированного запроса.

**Синтаксис оператора CreateTable:**

Create Table имя\_таблицы (<определение\_столбца> [, <определение\_столбца>]

[, primary key (<список\_столбцов\_первичного\_ключа>)]

[, foreign key (<список\_столбцов\_внешнего\_ключа>)

Reference <имя\_родительской\_таблицы> ] ), где

1. <определение\_столбца> - включает в себя имя столбца и его тип, который пишется через пробел после имени;

2. секция primary key (<список\_столбцов\_первичного\_ключа>) – задает первичный ключ таблицы;

3. foreign key (<список\_столбцов\_внешнего\_ключа>)

Reference <имя\_родительской\_таблицы> - задает ссылочную целостность между таблицами.

Например: создать базу данных «Учет товаров», состоящую из 4-х таблиц:

1. таблица «Поставщики» - S (Snum, Sname, Status, City), первичный ключ Snum;

2. таблица «Товары» - T (Tnum, Tname, Cena, EdIzm, City), первичный ключ Tnum;

3. таблица «Организации» - D (Dnum, Dname, City), первичный ключ Dnum;

4. таблица «Поставки» - STD (Snum, Tnum, Dnum, Kol\_vo, Data), первичный ключ Snum, Tnum, Dnum, причем таблица «Поставки» связана с первыми тремя таблицами по полям Snum, Tnum, Dnum.

Решение:

Create Table S (Snum nvarchar (4),

Sname nvarchar (20),

Status integer,

City nvarchar (20),

Primary key (Snum));

Create Table T (Tnum nvarchar (4),

Tname nvarchar (15),

Cena decimal(18,2),

EdIzm nvarchar (10),

City nvarchar (20),

Primary key (Tnum));

Create Table D (Dnum nvarchar (4),

Dname nvarchar (20),

City nvarchar (20),

Primary key (Dnum));

Create Table STD (Snum nvarchar (4),

Tnum nvarchar (4),

Dnum nvarchar (4),

Kol\_vo integer,

Data nvarchar (8),

Primary key (Snum, Pnum, Dnum),

Foreign key (Snum) references S,

Foreign key (Pnum) references P,

Foreign key (Dnum) references D

)

**Типы данных T-SQL**

При создании таблицы для всех ее столбцов необходимо указать определенный тип данных. Тип данных определяет, какие значения могут храниться в столбце, сколько они будут занимать места в памяти.

Язык T-SQL предоставляет множество различных типов. В зависимости от характера значений все их можно разделить на группы.

**Числовые типы данных**

* BIT: хранит значение 0 или 1. Фактически является аналогом булевого типа в языках программирования. Занимает 1 байт.
* TINYINT: хранит числа от 0 до 255. Занимает 1 байт. Хорошо подходит для хранения небольших чисел.
* SMALLINT: хранит числа от –32 768 до 32 767. Занимает 2 байта
* INT: хранит числа от –2 147 483 648 до 2 147 483 647. Занимает 4 байта. Наиболее используемый тип для хранения чисел.
* BIGINT: хранит очень большие числа от -9 223 372 036 854 775 808 до 9 223 372 036 854 775 807, которые занимают в памяти 8 байт.
* DECIMAL: хранит числа c фиксированной точностью. Занимает от 5 до 17 байт в зависимости от количества чисел после запятой.

Данный тип может принимать два параметра precision и scale: DECIMAL(precision, scale).

Параметр precision представляет максимальное количество цифр, которые может хранить число. Это значение должно находиться в диапазоне от 1 до 38. По умолчанию оно равно 18.

Параметр scale представляет максимальное количество цифр, которые может содержать число после запятой. Это значение должно находиться в диапазоне от 0 до значения параметра precision. По умолчанию оно равно 0.

* NUMERIC: данный тип аналогичен типу DECIMAL.
* SMALLMONEY: хранит дробные значения от -214 748.3648 до 214 748.3647. Предназначено для хранения денежных величин. Занимает 4 байта. Эквивалентен типу DECIMAL(10,4).
* MONEY: хранит дробные значения от -922 337 203 685 477.5808 до 922 337 203 685 477.5807. Представляет денежные величины и занимает 8 байт. Эквивалентен типу DECIMAL(19,4).
* FLOAT: хранит числа от –1.79E+308 до 1.79E+308. Занимает от 4 до 8 байт в зависимости от дробной части.

Может иметь форму определения в виде FLOAT(n), где n представляет число бит, которые используются для хранения десятичной части числа (мантиссы). По умолчанию n = 53.

* REAL: хранит числа от –340E+38 to 3.40E+38. Занимает 4 байта. Эквивалентен типу FLOAT(24).

Примеры числовых столбцов:

Salary MONEY,

TotalWeight DECIMAL(9,2),

Age INT,

Surplus FLOAT

**Типы данных, представляющие дату и время**

* DATE: хранит даты от 0001-01-01 (1 января 0001 года) до 9999-12-31 (31 декабря 9999 года). Занимает 3 байта.
* TIME: хранит время в диапазоне от 00:00:00.0000000 до 23:59:59.9999999. Занимает от 3 до 5 байт.

Может иметь форму TIME(n), где n представляет количество цифр от 0 до 7 в дробной части секунд.

* DATETIME: хранит даты и время от 01/01/1753 до 31/12/9999. Занимает 8 байт.
* DATETIME2: хранит даты и время в диапазоне от 01/01/0001 00:00:00.0000000 до 31/12/9999 23:59:59.9999999. Занимает от 6 до 8 байт в зависимости от точности времени.

Может иметь форму DATETIME2(n), где n представляет количество цифр от 0 до 7 в дробной части секунд.

* SMALLDATETIME: хранит даты и время в диапазоне от 01/01/1900 до 06/06/2079, то есть ближайшие даты. Занимает от 4 байта.
* DATETIMEOFFSET: хранит даты и время в диапазоне от 0001-01-01 до 9999-12-31. Сохраняет детальную информацию о времени с точностью до 100 наносекунд. Занимает 10 байт.

Распространенные форматы дат:

* yyyy-mm-dd - 2017-07-12
* dd/mm/yyyy - 12/07/2017
* mm-dd-yy - 07-12-17

В таком формате двузначные числа от 00 до 49 воспринимаются как даты в диапазоне 2000-2049. А числа от 50 до 99 как диапазон чисел 1950 - 1999.

* Month dd, yyyy - July 12, 2017

Распространенные форматы времени:

* hh:mi - 13:21
* hh:mi am/pm - 1:21 pm
* hh:mi:ss - 1:21:34
* hh:mi:ss:mmm - 1:21:34:12
* hh:mi:ss:nnnnnnn - 1:21:34:1234567

**Строковые типы данных**

* CHAR: хранит строку длиной от 1 до 8 000 символов. На каждый символ выделяет по 1 байту. Не подходит для многих языков, так как хранит символы не в кодировке Unicode.

Количество символов, которое может хранить столбец, передается в скобках. Например, для столбца с типом CHAR(10) будет выделено 10 байт. И если мы сохраним в столбце строку менее 10 символов, то она будет дополнена пробелами.

* VARCHAR: хранит строку. На каждый символ выделяется 1 байт. Можно указать конкретную длину для столбца - от 1 до 8 000 символов, например, VARCHAR(10). Если строка должна иметь больше 8000 символов, то задается размер MAX, а на хранение строки может выделяться до 2 Гб: VARCHAR(MAX).

Не подходит для многих языков, так как хранит символы не в кодировке Unicode.

В отличие от типа CHAR если в столбец с типом VARCHAR(10) будет сохранена строка в 5 символов, то в столце будет сохранено именно пять символов.

* NCHAR: хранит строку в кодировке Unicode длиной от 1 до 4 000 символов. На каждый символ выделяется 2 байта. Например, NCHAR(15)
* NVARCHAR: хранит строку в кодировке Unicode. На каждый символ выделяется 2 байта.Можно задать конкретный размер от 1 до 4 000 символов: . Если строка должна иметь больше 4000 символов, то задается размер MAX, а на хранение строки может выделяться до 2 Гб.

Еще два типа TEXT и NTEXT являются устаревшими и поэтому их не рекомендуется использовать. Вместо них применяются VARCHAR и NVARCHAR соответственно.

Примеры определения строковых столбцов:

Email VARCHAR(30),

Comment NVARCHAR(MAX)

**Бинарные типы данных**

* BINARY: хранит бинарные данные в виде последовательности от 1 до 8 000 байт.
* VARBINARY: хранит бинарные данные в виде последовательности от 1 до 8 000 байт, либо до 2^31–1 байт при использовании значения MAX (VARBINARY(MAX)).

Еще один бинарный тип - тип IMAGE является устаревшим, и вместо него рекомендуется применять тип VARBINARY.

**Остальные типы данных**

* UNIQUEIDENTIFIER: уникальный идентификатор GUID (по сути строка с уникальным значением), который занимает 16 байт.
* TIMESTAMP: некоторое число, которое хранит номер версии строки в таблице. Занимает 8 байт.
* CURSOR: представляет набор строк.
* HIERARCHYID: представляет позицию в иерархии.
* SQL\_VARIANT: может хранить данные любого другого типа данных T-SQL.
* XML: хранит документы XML или фрагменты документов XML. Занимает в памяти до 2 Гб.
* TABLE: представляет определение таблицы.
* GEOGRAPHY: хранит географические данные, такие как широта и долгота.
* GEOMETRY: хранит координаты местонахождения на плоскости.