



**УПРАВЛЕНИЕ
РЕЛЯЦИОННОЙ
БАЗОЙ ДАННЫХ С
ПОМОЩЬЮ SQL**

В данной работе рассматривается диалект Transact-SQL, используемый в СУБД Microsoft SQL Server. Он базируется на стандарте SQL-92. Transact-SQL добавляет к базовому языку SQL некоторые дополнительные операторы, расширяющие его возможности в работе с данными.

Подобно всем другим языкам программирования Transact-SQL представляет собой набор инструкций (или операторов), все множество которых можно разбить на следующие группы:

- комментарии;
- алфавит;
- идентификаторы;
- выражения;
- ключевые слова;
- операторы;
- функции.

Комментарий

Комментарий – это текстовая строка, которая игнорируется при выполнении программы и служит для пояснения выполняемых действий.

Комментарии можно определить двумя способами:

- с помощью символов `/*` (начало комментария) и `*/` (конец комментария), между которыми можно разместить любое количество строк комментария;
- с помощью двух символов `--`, за которыми можно разместить одну строку комментария.

Алфавит

В Transact-SQL используются:

- ✓ символы латинского алфавита,
- ✓ цифры,
- ✓ символы подчеркивания (_),
- ✓ процента (%),
- ✓ звездочка (*),
- ✓ вопросительный знак (?),
- ✓ восклицательный знак (!),
- ✓ подстановочный знак (#),
- ✓ знаки арифметических операций.

В качестве разделителей
используется:

- ✓ двойная кавычка (“”),
- ✓ апостроф (‘),
- ✓ запятая (,),
- ✓ точка (.),
- ✓ точка с запятой (;),
- ✓ двоеточие (:),
- ✓ квадратные и круглые скобки.

Запятые используются для разделения элементов списка, например, имен столбцов таблицы:

ИМЯ, АДРЕС, ГОРОД.

Квадратные скобки используются для задания имен столбцов, которые содержат недопустимые символы, включая пробелы и разделители. Часто имя столбца таблицы может быть образовано из нескольких слов [Название компании].

Если в запрос включены поля нескольких таблиц, то для разделения имени таблицы и имени поля используется точка, например:

Фирма.Адрес

Строки можно заключать как в одиночные, так и двойные кавычки.

Идентификаторы

Идентификаторы используются для ссылки на объекты баз данных. Построение идентификаторов выполняется на основе следующих правил:

- число символов не более 128, причем в качестве первого может использоваться только буква, символ подчеркивания и знаки @, #, далее могут следовать буквы, цифры или символы #, @, \$ и символ подчеркивания;
- идентификатор, начинающийся с символа @, указывает на локальную переменную; два первых символа @@ говорят о том, что идентификатор указывает на глобальную переменную; символ # в начале идентификатора указывает на то, что идентификатор, является именем временной таблицы или процедуры;
- хотя пробелы в идентификаторах запрещены, это ограничение можно обойти, заключив идентификатор в квадратные скобки или двойные кавычки.

Ключевые слова

Ключевые слова – predetermined идентификаторы, зарезервированные для определенных целей. Ключевые слова используются для обозначения команд, инструкций, предложений, операторов, функций.

Операторы

Операторы Transact-SQL можно разбить на четыре категории:

- арифметические операторы;
- операторы сравнения;
- оператор для слияния строк;
- логические операторы.

Арифметические операторы

Арифметические операторы применимы только к **числовым значениям** и должны иметь **два числовых операнда**. Исключение составляет знак минус (-), изменяющий знак операнда. В этом случае минус называется *унарным минусом*.

Оператор	Описание
+	Суммирование двух операндов
-	Определение разности двух операндов
-	Изменение знака операнда
*	Перемножение двух операндов
/	Деление первого операнда на второй операнд
%	Определение остатка целочисленного деления. Например, 14 % 4 возвращает 2

Операторы сравнения

Эти операторы используются для сравнения символов, чисел, дат и возвращают значение **TRUE** (истина) или **FALSE** (ложь) в зависимости от результатов сравнения. Исключением является случай, когда один из операндов имеет значение **NULL**. В этом случае любое сравнение возвращает значение **NULL**.

Оператор	Описание	Пример	Результат
<	Меньше	10 < 55	TRUE
		10 < NULL	NULL
<=	Меньше или равно	4 <= 9	TRUE
=	Равно	2 = 3	FALSE
>=	Больше или равно	2 >= 3	FALSE
>	Больше	33 > 12	TRUE
<>	Не равно	2 <> 5	TRUE

Логические операторы

Операторы этой группы применимы только к целочисленным типам данных: `int`, `smallint`, `tinyint`, `bit` и трактуют целые значения как последовательность двоичных значений 0 и 1.

Оператор	Запись оператора в командах SQL	Описание
&	AND	Конъюнкция (логическое И)
	OR	Дизъюнкция (логическое ИЛИ)
~	NOT	Логическое отрицание
^		Исключающее ИЛИ

Оператор слияния строк (конкатенация). Конкатенация объединяет две строки символов в единую строку. Например, 'FDS' + 'ASD' дает строку 'FDSASD'.

Типы данных

Тип данных – это характеристика, которая задается для **столбца таблицы** или **переменной**. При этом определяется тип хранящейся в них информации.

Тип	объем	Описание
Char	1 байт на символ	Строка символов размером до 8 К. Если фактическая длина строки меньше n, то она дополняется справа пробелами.
Money	8 байт	Денежное выражение для числовой величины. Выводит число с четырьмя десятичными разрядами и установленным обозначением денежной единицы.
Datetime	8 байт	Дата и время. Позволяет хранить дату и время начиная с 01.01.1753 нашей эры и заканчивая 31.12.9999 нашей эры с точностью 3.33 мс.
Int	4 байта	Целое число от -2,147,483,648 до 2,147,483,647
Image	16 байт	Поле для ссылки на объект OLE
Text	16 байт	Поле примечаний для ссылки на блок данных. В полях типа Text хранится текстовая информация, размер которой кратен 8 Кб. Максимальный размер хранимой строки 2,147,483,647 символов.
Real	4 байта	Тип данных с плавающей точкой. Положительные или отрицательные числа можно сохранять с точностью до 7 цифр. Данные типа Real принадлежат интервалу от -3.4E+38 до +3.4E+38.

Агрегатные функции

Базовые (агрегатные) функции позволяют производить вычисления на основе данных, хранящихся в некотором столбце таблицы. Параметром агрегатной функции может быть имя столбца таблицы или звездочка (*).

Функция	Действие
AVG	Возвращает среднее арифметическое значение указанного столбца таблицы
COUNT	Возвращает число строк таблицы, в которых значение заданного столбца не равно значению NULL. Если параметром функции является звездочка, то подсчет ведется по всем строкам таблицы.
MAX	Возвращает максимальное значение в столбце
MIN	Возвращает минимальное значение в столбце
SUM	Возвращает сумму значений заданного столбца

Строковые функции

Функция	Результат
CHAR(целое число)	Преобразует целое число в символ.
CHARINDEX(символьное выражение, строка)	Возвращает номер позиции, с которой начинается символьное выражение в строке. Если заданное символьное выражение в строке отсутствует, то возвращается 0.
REVERSE(строка)	Возвращает строку «задом наперед».
RIGHT(строка, целое число)	Возвращает часть строки, начиная с указанной позиции.
SPACE(целое число)	Возвращает строку из указанного числа пробелов. Если указано отрицательное число, то возвращается пустая строка.
STR(число, число символов, количество десятичных знаков)	Преобразует число в строку символов.
SUBSTRING(строка, начальная позиция, длина)	Возвращает часть строки, указанной длины, от начальной позиции.
UPPER(строка)	Переводит символы из нижнего регистра в верхний.

Математические функции

Математические функции работают с числовыми типами данных. Значения, возвращаемые этими функциями, имеют шесть десятичных разрядов.

Функция	Типы аргументов	Результат
ABS	Число	Абсолютное значение.
COS, SIN, COT, TAN	Число с плавающей запятой, задающее угол в радианах	Косинус, синус, котангенс и тангенс угла.
CEILING	Число	Наименьшее целое, которое больше или равно указанному аргументу.
DEGREES	Число	Преобразует угол из радиан в градусы.
EXP	Число с плавающей запятой	Экспонента от аргумента.
FLOOR	Число	Наибольшее целое, которое меньше или равно указанному аргументу.
LOG	Число с плавающей запятой	Натуральный логарифм аргумента.
LOG10	Число с плавающей запятой	Десятичный логарифм аргумента.
PI	-	Возвращает константу 3.141592653 (число π)
POWER	Число, у (число)	Возвращает аргумент в степени у.
RADIANS	Число	Преобразует угол из градусов в радианы.
RAND	Целое число. Аргумент необязателен.	Возвращает случайное число с плавающей запятой в диапазоне от 0 до 1. Аргумент может использоваться в качестве начального значения.
ROUND	Число, количество цифр	Число округляется до указанного количества цифр после запятой.
SIGN	Число	Возвращает знак числа.
SQRT	Число с плавающей запятой	Квадратный корень от числа.

Функции для работы с датами

Функция	Тип аргумента	Результат
YEAR	Дата	Возвращает год в виде целого числа
MONTH	Дата	Возвращает номер месяца
DAY	Дата	Возвращает номер дня
GETDATE		Возвращает текущее время и дату.

Создание таблиц БД

CREATE TABLE

```
[<Имя базы данных>.[<Имя владельца>].]<Имя таблицы>  
( {<Определение столбца>  
| <Имя столбца> AS <Рассчитываемое выражение>  
| <Условия на значения таблицы> } [, ...n])  
[ON {<Группа> | DEFAULT}]  
[TEXTIMAGE_ON {Группа | DEFAULT}],
```

где столбцы определяются следующим образом:

```
<Имя столбца> [<Тип данных> [<Длина>]] [NULL | NOT NULL]  
[IDENTITY [(<Начало>, <Приращение>)]  
[NOT FOR REPLICATION]]  
[ROWGUIDCOL]
```

Пример создания таблицы ОПЛАТА в базе данных ГОСТИНИЦА

```
CREATE TABLE Гостиница.dbo.Оплата  
(Номер_оплаты Integer IDENTITY (1,1) PRIMARY KEY  
CLUSTERED,  
Код_клиента Integer NULL,  
Оплата Money DEFAULT 0,  
Дата_прибытия Datetime DEFAULT GETDATE())
```

Параметры команды

<Имя базы данных> - имя базы данных, которое можно не указывать, если база данных является текущей;

<Имя владельца> - владелец создаваемой таблицы;

<Имя таблицы> - уникальное для данной базы данных имя таблицы;

<Имя столбца> - уникальное имя столбца в таблице;

<Тип данных> - один из predetermined или созданных пользователем типов данных;

NULL | NOT NULL – ключевые слова, определяющие, разрешено или нет использование в столбце значение NULL;

PRIMARY KEY – первичный ключ;

FOREIGN KEY – внешний ключ;

REFERENCES – определение ссылки на другую таблицу;

NOT NULL – указывает, что значение атрибута не должно быть пустым;

Оператор создания БД

```
CREATE DATABASE имя_базы_данных;
```

```
CREATE DATABASE IF NOT EXISTS имя_базы_данных;
```

```
CONNECT имя_базы_данных
```

Описание таблиц и атрибутов предметной области "Университет"

Название таблицы	Название атрибута	Описание
<i>Disp</i>		Информация о дисциплине
	<i>ID_Disp</i>	Идентификатор дисциплины
	<i>Type</i>	Тип дисциплины (может принимать значения: ЛК – лекция, ПР – практическое занятие, ЛБ – лабораторные занятия, КР – курсовая работа)
	<i>Cafedra</i>	Название ведущей кафедры
	<i>Disp_Name</i>	Название дисциплины
	<i>Phone</i>	Телефон кафедры
<i>Person</i>		Информация о сотруднике университета
	<i>FIO</i>	Фамилия, имя, отчество сотрудника
	<i>Tab_N</i>	Табельный номер сотрудника
	<i>Ac_Degree</i>	Ученая степень сотрудника
<i>Teach_Load</i>		Нагрузка на сотрудника
	<i>Teach_ID</i>	Идентификатор преподавателя, ведущего дисциплину
	<i>Start_Work</i>	Время начала работы

Описание таблиц и атрибутов предметной области "Университет"

```
CREATE TABLE Disp(
  ID_Dispatch INTEGER not null primary key,
  Disp_Name CHAR(20) not null,
  Cafedra CHAR(50),
  Phone CHAR(8)
);

CREATE TABLE Person(
  FIO CHAR(30) not null,
  Tab_N INTEGER not null,
  Ac_Degree CHAR(20),
  PRIMARY KEY (Tab_N)
);

CREATE TABLE Teach_Load(
  Teach_ID INTEGER NOT NULL PRIMARY KEY,
  Start_Of_Work DATE DEFAULT 'NOW' not null,
  Person_ID INTEGER not null,
  Disp_ID INTEGER not null,
  FOREIGN KEY (Person_ID) REFERENCES Person (Tab_N),
  FOREIGN KEY (Disp_ID) REFERENCES Disp (ID_Dispatch)
);
```