**Обобщенные табличные выражения (ОТВ)**

Обобщенные табличные выражения (ОТВ) можно представить себе как временные результирующие наборы, определенные в области выполнения единичных инструкций SELECT, INSERT, UPDATE, DELETE или CREATE VIEW. ОТВ, как и производные таблицы, не сохраняются в базе данных в виде объектов, время их жизни ограничено продолжительностью запроса. Но, в отличие от производных таблиц, ОТВ могут ссылаться сами на себя, а на них один и тот же запрос может ссылаться несколько раз.

В основном ОТВ предназначены для:

1. Создания рекурсивных запросов.
2. Замены представлений в тех случаях, когда нет необходимости сохранять в метаданных базы его определение.
3. Многократных ссылок на результирующую таблицу из одной и той же инструкции.

ОТВ могут быть определены в пользовательских подпрограммах (функциях, хранимых процедурах, триггерах, представлениях).

Базовый синтаксиса ОТВ:

WITH имя\_ОТВ [ ( список\_имен\_столбцов ) ]

AS

( определение\_ОТВ )

Список имен столбцов необязателен только в том случае, если всем результирующим столбцам в определении запроса присвоены уникальные имена.

В простейшем случае инструкция для обращения к ОТВ имеет вид:

SELECT {\*|список\_имен\_столбцов} FROM имя\_ОТВ ;

**Пример.** Отобразить среднее количество поставок для поставщиков в учебной базе данных SPJ.

WITH CTE (SupplierNo, NumberOfShips)

AS

(

SELECT Sno, COUNT(\*) AS Total

FROM SPJ

WHERE Sno IS NOT NULL

GROUP BY Sno

)

SELECT AVG(NumberOfShips) AS "Среднее количество поставок для поставщиков"

FROM CTE;

GO

**Рекомендации по созданию и использованию обобщенных табличных выражений**

* Определение ОТВ задается инструкциями SELECT, результирующий набор которых заполняет ОТВ. Инструкции SELECT для определения ОТВ должны удовлетворять таким же требованиям, что и при создании представления, за исключением того, что ОТВ не может определять другое ОТВ.
* Если в определении ОТВ задано несколько инструкций SELECT, то они должны быть соединены одним из следующих операторов работы с наборами: UNION ALL, UNION, EXCEPT или INTERSECT.
* За ОТВ должна следовать одиночная инструкция SELECT, INSERT, UPDATE, MERGE или DELETE, ссылающаяся на некоторые или на все столбцы ОТВ. ОТВ может задаваться также в инструкции CREATE VIEW как часть определяющей инструкции SELECT представления.
* ОТВ могут иметь ссылки сами на себя, а также на ОТВ, определенные до этого в том же предложении WITH. Ссылки на определяемые далее ОТВ недопустимы.
* Задание в одном ОТВ нескольких предложений WITH недопустимо. Например, если определение ОТВ содержит вложенный запрос, этот вложенный запрос не может содержать вложенное предложение WITH, определяющее другое ОТВ.
* Следующие предложения не могут использоваться в определении ОТВ:
  + COMPUTE или COMPUTE BY
  + ORDER BY (за исключением случаев задания предложения TOP)
  + INTO
  + Предложение OPTION с подсказками в запросе
  + FOR XML
  + FOR BROWSE
* Если ОТВ используется в инструкции, являющейся частью пакета, за инструкцией, стоящей перед ней, должен следовать символ точки с запятой.
* Запрос, ссылающийся на ОТВ, может использоваться для определения курсора.
* В ОТВ могут быть ссылки на таблицы, находящиеся на удаленных серверах.

**Рекурсивные запросы, использующие ОТВ**

В ранних версиях SQL Server рекурсивный запрос моделировался с использованием временных таблиц и курсоров. Начиная с версии SQL Server 2005 рекурсивный запрос реализуется с помощью ОТВ, которое ссылается само на себя, создавая рекурсивное ОТВ. Таким образом, запрос именуется рекурсивным, если он ссылается на рекурсивное ОТВ.

Обычно рекурсивные запросы используются для возвращения иерархических данных, например, отображения сотрудников в структуре организации или данных в сценарии ведомости материалов, в котором родительский продукт состоит из одного или более компонентов, и эти компоненты могут, в свою очередь, состоять из вспомогательных компонентов или являться компонентами других родителей.

**Структура рекурсивного ОТВ**

Рекурсивное ОТВ должно содержать минимум один закрепленный элемент и один рекурсивный элемент. Следующий псевдокод отображает компоненты простого рекурсивного ОТВ, которое содержит один закрепленный элемент и один рекурсивный элемент.

WITH имя\_ОТВ [(список\_имен\_столбцов )]

AS

(

-- Определение закрепленного элемента

UNION ALL

-- Определение рекурсивного элемента

)

-- Инструкция, использующая ОТВ

SELECT {\*|список\_имен\_столбцов} FROM имя\_ОТВ

Рекурсивное выполнение имеет следующую семантику:

* разбиение ОТВ на закрепленный и рекурсивный элементы;
* запуск закрепленного элемента с созданием первого вызова или базового результирующего набора (T0);
* запуск рекурсивного элемента, где Ti — это вход, а Ti+1 — это выход;
* повторение шага 3 до тех пор, пока не вернется пустой набор;
* возвращение результирующего набора. Результирующий набор получается с помощью инструкции UNION ALL от T0 до Tn.

**Пример**. Получить список служащих вымышленной торговой компании, начиная с высшего должностного лица.

-- Создание таблицы.

CREATE TABLE dbo.MyEmployees

(

EmployeeID smallint NOT NULL,

FirstName nvarchar(30) NOT NULL,

LastName nvarchar(40) NOT NULL,

Title nvarchar(50) NOT NULL,

DeptID smallint NOT NULL,

ManagerID int NULL,

CONSTRAINT PK\_EmployeeID PRIMARY KEY CLUSTERED (EmployeeID ASC)

) ;

-- Заполнение таблицы значениями.

INSERT INTO dbo.MyEmployees VALUES (1, N'Иван', N'Петров', N'Главный исполнительный директор',16,NULL) ;

INSERT INTO dbo.MyEmployees VALUES (273, N'Борис', N'Иванов', N'Вице-президент по продажам',3,1) ;

INSERT INTO dbo.MyEmployees VALUES (274, N'Степан', N'Гаврилов', N'Менеджер по продажам в Европе',3,273) ;

INSERT INTO dbo.MyEmployees VALUES (275, N'Михаил', N'Глинка', N'Торговый представитель',3,274) ;

INSERT INTO dbo.MyEmployees VALUES (276, N'Елена', N'Пчелкина', N'Торговый представитель',3,274) ;

INSERT INTO dbo.MyEmployees VALUES (285, N'Сергей', N'Фролов', N'Менеджер по продажам в Азии',3,273) ;

INSERT INTO dbo.MyEmployees VALUES (286, N'Леонид', N'Дискин', N'Торговый представитель',3,285) ;

INSERT INTO dbo.MyEmployees VALUES (16, N'Денис',N'Ершов', N'Менеджер по маркетингу', 4, 273) ;

INSERT INTO dbo.MyEmployees VALUES (23, N'Мария', N'Ушакова', N'Специалист по маркетингу', 4, 16) ;

GO

-- Определение ОТВ

WITH DirectReports (ManagerID, EmployeeID, Title, DeptID, Level)

AS

(

-- Определение закрепленного элемента

SELECT e.ManagerID, e.EmployeeID, e.Title, e.DeptID, 0 AS Level

FROM dbo.MyEmployees AS e

WHERE ManagerID IS NULL

UNION ALL

-- Определение рекурсивного элемента

SELECT e.ManagerID, e.EmployeeID, e.Title, e.DeptID, Level + 1

FROM dbo.MyEmployees AS e INNER JOIN DirectReports AS d ON e.ManagerID = d.EmployeeID

)

-- Инструкция, использующая ОТВ

SELECT ManagerID, EmployeeID, Title, DeptID, Level

FROM DirectReports ;

GO

**Анализ кода примера**

* Рекурсивное ОТВ DirectReports определяет один закрепленный элемент и один рекурсивный элемент.
* Закрепленный элемент возвращает базовый результирующий набор T0. Это самое главное должностное лицо компании; значит, этот служащий не отчитывается перед управляющим. Ниже приведен результирующий набор, возвращенный закрепленным элементом.

NULL 1 Главный исполнительный директор 16 0

* Рекурсивный элемент возвращает прямых подчиненных служащего в результирующий набор закрепленного элемента. Это получается при соединении таблицы MyEmployees и ОТВ DirectReports. Это ссылка на само ОТВ, которое устанавливает рекурсивный вызов. В зависимости от служащего в ОТВ DirectReports в качестве входа (Ti) соединение (MyEmployees.ManagerID = DirectReports.EmployeeID) возвращает выход (Ti+1) — это служащие, руководителем которых является (Ti). Таким образом, первый шаг цикла рекурсивного элемента возвращает следующий

результирующий набор:

1 273 Вице-президент по продажам 3 1

* Рекурсивный элемент постоянно активируется. Второй шаг цикла рекурсивного элемента использует однострочный результирующий набор в шаге 3 (содержащий EmployeeID 273) в качестве входного значения и возвращает следующий результирующий набор:

273 16 Менеджер по маркетингу 4 2

273 274 Менеджер по продажам в Европе 3 2

273 285 Менеджер по продажам в Азии 3 2

Третий шаг цикла рекурсивного элемента использует вышеупомянутый результирующий набор в качестве входного значения и возвращает следующий результирующий набор:

285 286 Торговый представитель 3 3

274 275 Торговый представитель 3 3

274 276 Торговый представитель 3 3

16 23 Специалист по маркетингу 4 3

* Конечный результирующий набор, возвращенный запущенным запросом, представляет собой объединение всех результирующих наборов, сформированных закрепленным и рекурсивным элементами. Ниже приведен полный результирующий набор, возвращенный примером.

NULL 1 Главный исполнительный директор 16 0

1 273 Вице-президент по продажам 3 1

273 16 Менеджер по маркетингу 4 2

273 274 Менеджер по продажам в Европе 3 2

273 285 Менеджер по продажам в Азии 3 2

285 286 Торговый представитель 3 3

274 275 Торговый представитель 3 3

274 276 Торговый представитель 3 3

16 23 Специалист по маркетингу 4 3

Количество возвращаемых уровней можно ограничивать с помощью WHERE-предложения, например, так

SELECT \* FROM DirectReports WHERE Level <= 2 ;

Неправильно составленное рекурсивное ОТВ может привести к бесконечному циклу. При тестировании результатов рекурсивного запроса можно ограничить число уровней рекурсии, используя подсказку MAXRECURSION, например, так

SELECT \* FROM DirectReports OPTION (MAXRECURSION 2) ;

После исправления ошибки в коде подсказка MAXRECURSION больше не нужна.

**Рекомендации по созданию и использованию рекурсивных ОТВ**

* Определение рекурсивного ОТВ должно содержать по крайней мере два запроса: первый запрос называется закрепленным элементом, а второй запрос называется рекурсивным элементом. Могут быть определены несколько закрепленных элементов и несколько рекурсивных элементов, однако все закрепленные элементы должны быть поставлены перед первым рекурсивным элементом. Все запросы в определении ОТВ являются закрепленными элементами, если только они не ссылаются на само ОТВ.
* Закрепленные элементы должны объединяться одним из следующих операторов работы с наборами: UNION ALL, UNION, INTERSECT или EXCEPT. UNION ALL является единственным оператором, который может находиться между последним закрепленным элементом и первым рекурсивным элементом, а также при объединении нескольких рекурсивных элементов.
* Типы и количество столбцов закрепленных и рекурсивных элементов должны совпадать.
* Предложение FROM рекурсивного элемента должно ссылаться на ОТВ только один раз.
* Следующие элементы недопустимы в определении рекурсивного элемента:
  + SELECT DISTINCT
  + GROUP BY
  + HAVING
  + Скалярный агрегат
  + TOP
  + LEFT, RIGHT, OUTER JOIN (INNER JOIN допускается)
  + Вложенные запросы
  + Подсказка, применимая к рекурсивной ссылке на ОТВ в определении ОТВ
* Все столбцы, возвращаемые рекурсивным ОТВ, могут содержать значения NULL, независимо от того, могут ли иметь значения NULL столбцы, возвращаемые участвующими инструкциями SELECT.
* Неправильно составленное рекурсивное ОТВ может привести к бесконечному циклу. Для предотвращения бесконечного цикла можно ограничить количество уровней рекурсии, допустимых для определенной инструкции, с помощью подсказки MAXRECURSION и значения в диапазоне от 0 до 32 767 в предложении OPTION инструкции INSERT, UPDATE, MERGE, DELETE или SELECT. Серверное значение по умолчанию равно 100. Если указано значение 0, ограничения не применяются. В одной инструкции может быть указано только одно значение MAXRECURSION.
* Представление, содержащее рекурсивное ОТВ, не может использоваться для обновления данных.
* Курсоры могут определяться на запросах при помощи ОТВ. Для рекурсивных ОТВ допустимы только однонаправленные и статические курсоры (курсоры моментального снимка). Если в рекурсивном ОТВ указан курсор другого типа, тип курсора преобразуется в статический.