

## ЛАБОРАТОРНАЯ РАБОТА №13

*Тема:* Обобщенные табличные выражения (CTE). Разработка SQL рекурсивных запросов на основе внутренних представлений.

*Цель:* Пробрести навыки составления рекурсивных SQL запросов использующих внутренние представления.

В данной лабораторной работе используются база данных HR, созданная в рамках первой лабораторной работы. Также используется таблица TB\_ELEKTROSTAL\_2018, которая создана и наполнена данными в процессе выполнения лаб. работы №1.

Дополнительно используется удаленная база данных проекта stackoverflow.com. Структура этой базы данных описана в методических указаниях к выполнению лабораторной работы № 3. Напомним, адрес ресурса <http://data.stackexchange.com/stackoverflow/query/new>.

Обобщенное табличное выражение (CTE) имеет значительное преимущество, так как оно может ссылаться на себя, создавая рекурсивное CTE. Рекурсивное CTE является выражением, в котором начальное CTE многократно выполняется, чтобы возвращать подмножество данных до тех пор, пока не получится конечный результирующий набор.

В SQL запрос именуется рекурсивным, если он ссылается на рекурсивное CTE. Обычно рекурсивные запросы используются для возвращения иерархических данных, например: отображение сотрудников в структуре организации.

Рекурсивное CTE может существенно упростить код, требуемый для запуска рекурсивного запроса в рамках инструкций SELECT, INSERT, UPDATE, DELETE или CREATE VIEW. В более ранних версиях SQL, чтобы контролировать поток рекурсивных шагов, рекурсивный запрос обычно требует использование временных таблиц, курсоров и логики.

### **Структура рекурсивного CTE**

Структура рекурсивного CTE в SQL аналогична рекурсивным процедурам в других языках программирования. Но рекурсивные процедуры в других языках возвращают скалярное значение, а рекурсивное CTE может возвращать несколько строк.

Рекурсивное CTE состоит из трех элементов.

1. Вызов процедуры.

Первый вызов рекурсивного CTE состоит из одного или более параметров *CTE\_query\_definitions*, соединенных операторами UNION ALL, UNION, EXCEPT или INTERSECT. Так как данные определения запроса формируют базовый результирующий набор структуры CTE, они называются закрепленными элементами.

Параметры *CTE\_query\_definitions* считаются закрепленными элементами, если они не ссылаются на само CTE. Все определения запросов закрепленных элементов должны размещаться перед определением первого рекурсивного элемента, а оператор UNION ALL должен использоваться для соединения последнего закрепленного элемента с первым рекурсивным элементом.

2. Рекурсивный вызов процедуры.

Рекурсивный вызов включает в себя от одного или более параметров *CTE\_query\_definitions*, которые соединены операторами UNION ALL, ссылающимися на само CTE. Данные определения запросов называются рекурсивными элементами.

3. Проверка завершения.

Проверка завершения происходит неявно; рекурсия останавливается, если из предыдущего вызова не вернулась ни одна строка.

Неправильно составленное рекурсивное CTE может привести к бесконечному циклу. Например, если запрос рекурсивного элемента возвращает одинаковые значения для родительского столбца и столбца потомка, то образуется бесконечный цикл.



```
SELECT n FROM numbers;          -- основной запрос
```

Следующий пример показывает семантику структуры рекурсивного CTE, возвращая иерархический список служащих компании, начиная с высшего должностного лица.

```
with p(name,id,lv1)  
as (  
    select first_name||' '||last_name,employee_id,0 lv1  
from employees where manager_id is null  
union all  
    select rpad('||',4*(p.lv1+1),'-  
' )||employees.first_name||'  
' ||employees.last_name,employees.employee_id,p.lv1+1  
from employees inner join p on  
employees.manager_id=p.id  
)  
SEARCH DEPTH FIRST BY id DESC SET ordinal  
select * from p
```

#### Анализ примера кода

1. Рекурсивное CTE `p` определяет один опорный элемент и один рекурсивный элемент.
2. Опорный элемент возвращает базовый результирующий набор  $T_0$ . Это самое главное должностное лицо компании; значит, этот служащий не отчитывается перед управляющим.
3. Рекурсивный элемент возвращает прямых подчиненных служащего в результирующий набор опорного элемента. Это получается при соединении таблицы `Employee` и `p` CTE. Это ссылка на само CTE, которое устанавливает рекурсивный вызов. В зависимости от служащего в CTE `p` в качестве входа ( $T_i$ ), соединение (`Employee.Manager_ID = p.ID`) возвращает выход ( $T_{i+1}$ ) — это служащие, чьим управляющим является ( $T_i$ ).
4. Рекурсивный элемент постоянно активизируется. Данный процесс повторяется до тех пор, пока рекурсивный элемент не вернет пустой результирующий набор.

5. Конечный результирующий набор, возвращенный запущенным запросом, представляет собой объединение всех результирующих наборов, сформированных закрепленным и рекурсивным элементами.

Обратите внимание на поле **lvl**. Это поле позволяет отследить уровень вложенности текущей итерации выполнения запроса. Также обратите внимание на предложение **search depth first ...**. Данная конструкция обеспечивает «поиск в глубину».

Для предупреждения заикливания вычислений можно использовать специальное указание **CYCLE**, где следует указать перечень (в общем случае) столбцов для распознавания хождения по кругу, придумать название столбца- индикатора (он автоматически включается в конечный ответ) и задать пару символов: для обозначения незаикленной строки и для обозначения строки, где было зафиксировано повторение значений в различительных столбцах. Например,

```
CYCLE id SET cyclemark TO 'X' DEFAULT '-'
```

### Порядок выполнения работы

Составить следующие запросы

- Составить рекурсивный запрос к БД HR. Результат запроса должен содержать две колонки. Первая колонка полное имя работника (поля `first_name` и `last_name`, разделенные пробелом). Вторая строка – латинская буква. Для каждого работника латинские буквы следуют от “A” до “Z”. Таким образом, строка каждого работника повторяется столько раз, сколько букв в английском языке.
- Составить рекурсивный запрос, который выводит числа от 1 до N. Здесь N – количество дней в текущем месяце.

### Содержание отчета:

1. Тема, цель лабораторной работы.

2. Примеры выполнения запросов к базе данных.
3. Составленные согласно заданию запросы и скриншоты полученных результатов.
5. Выводы.

**Контрольные вопросы:**

1. Опишите назначение рекурсивных запросов
2. Дайте описание синтаксической конструкции для описании рекурсивного запроса.
3. Каким ключевым выражением объединяются опорная часть запроса и его рекурсивная часть?
4. Выполните пример, приведенный в данных методических указаниях без конструкции `search depth first` и с этой конструкцией. Объясните разницу в результатах
5. Какая конструкция предназначена для предотвращения заикливания в рекурсивных запросах?
- 6.

Файл: Лабораторная работа 13  
Каталог: C:\Users\sss\Documents  
Шаблон: C:\Users\sss\AppData\Roaming\Microsoft\Шаблоны\Normal.dotm  
Заголовок:  
Содержание:  
Автор: sss  
Ключевые слова:  
Заметки:  
Дата создания: 17.11.2018 22:19:00  
Число сохранений: 14  
Дата сохранения: 19.11.2018 20:48:00  
Сохранил: sss  
Полное время правки: 219 мин.  
Дата печати: 19.11.2018 20:49:00  
При последней печати  
    страниц: 6  
    слов: 1 211 (прибл.)  
    знаков: 6 909 (прибл.)