

Практическая работа 5

Цель работы: использование результатов выборки как источник данных, использование обобщенного табличного выражения (ОТВ).

В подавляющем большинстве случаев операция выборки предполагает, что источником данных будут служить таблицы. В некоторых случаях это создает определенные неудобства, в частности, не позволяет получить выборку из результата выполнения предыдущего запроса. Решить эту задачу можно несколькими способами.

Представление. Представление является именованным результатом выполнения запроса. Поскольку представление – это объект базы данных, то для его создания и использования требуются дополнительные права.

Альтернативным вариантом, не требующим дополнительных прав, является использование запросов как источников данных или обобщенных табличных выражений.

Использование запроса как источника данных подразумевает, что пользователь создает запрос, дает ему псевдоним и использует этот псевдоним аналогично имени таблицы.

Рассмотрим сложный и объемный пример. Необходимо найти для каждой подкатегории количество товаров, у которых цена выше средней цены в подкатегории, и количество товаров, у которых цена ниже средней цены в подкатегории. Оформить вывод в виде трех столбцов: номер подкатегории, первый показатель, второй показатель. Решить подобную задачу традиционным способом достаточно сложно, но она относительно легко решается с использованием псевдонимов для результата выполнения запроса.

```
SELECT T1.PS, T1.c, T2.c FROM
(SELECT COUNT(*) AS c, [ProductSubcategoryID] AS PS
FROM [Production].[Product] AS P
WHERE [ListPrice]<(SELECT avg([ListPrice])
FROM [Production].[Product] AS PT
WHERE pt.ProductSubcategoryID=p.ProductSubcategoryID)
GROUP BY [ProductSubcategoryID]) AS T1 INNER JOIN
(SELECT count(*) AS c, [ProductSubcategoryID] AS PS
FROM [Production].[Product] AS P
WHERE [ListPrice]>=(SELECT avg([ListPrice])
FROM [Production].[Product] AS PT
WHERE pt.ProductSubcategoryID=p.ProductSubcategoryID)
GROUP BY [ProductSubcategoryID]) AS T2 ON
```

T1.PS=T2.PS

В данном примере созданы два независимых запроса, один из которых находит номер категории и количество товаров, цена у которых ниже средней цены в той же категории, к которой он относится, второй – такой же, но знак заменен на «больше или равно». Каждый запрос получил свой псевдоним, T1 и T2 соответственно, и каждому столбцу в запросе дан свой псевдоним. Запрос верхнего уровня использует запросы T1 и T2 точно так же, как обычные таблицы, проводит операцию INNER JOIN и выводит искомый результат.

К сожалению, использование такого подхода удобно, когда запрос, формирующий источник данных, не очень длинный. В противном случае читаемость кода резко падает, и в этом случае удобнее использовать обобщенное табличное выражение. Также обобщенное табличное выражение можно использовать при выполнении однократной операции модификации данных.

Общий синтаксис:

```
WITH <имя ОТВ>
( столбец 1, столбец 2, ... ]
AS
(запрос)
```

В нерекурсивных ОТВ недопустимо использовать операцию упорядочивания, за исключением случаев использования инструкции TOP.

Рассмотрим пример, в котором необходимо найти количество чеков, приходящихся на одного покупателя на каждый год.

```
WITH Sales_CTE (SalesPersonID, SalesORDERID, SalesYear)
AS
(
    SELECT SalesPersonID, SalesORDERID, YEAR(ORDERDate) AS
SalesYear
    FROM Sales.SalesORDERHeader
    WHERE SalesPersonID IS NOT NULL
)
SELECT SalesPersonID, COUNT(SalesORDERID) AS TotalSales, SalesYear
FROM Sales_CTE
GROUP BY SalesYear, SalesPersonID
ORDER BY SalesPersonID, SalesYear
```

В данном примере определено ОТВ с именем Sales_CTE, которое использовано как источник данных для запроса.

Если необходимо использовать несколько ОТВ в одном запросе, то их можно определить следующим образом.

```
WITH <имя ОТВ1>
( столбец 1, столбец 2, ... ]
AS
(запрос),
<имя ОТВ2>
( столбец 1, столбец 2, ... ]
AS
(запрос)
```

Примеры запросов с решениями

1 Найти покупателя, который каждый раз имел разный список товаров в чеке (по номенклатуре)

```
SELECT tmp.c
FROM
(SELECT soh.CustomerID AS c
, soh.SalesORDERID AS o
, CHECKSUM_AGG(sod.ProductID) AS ch
FROM [Sales].[SalesORDERDetail] AS sod JOIN
[Sales].[SalesORDERHeader] AS soh
ON sod.SalesORDERID=soh.SalesORDERID
GROUP BY soh.CustomerID, soh.SalesORDERID) tmp
GROUP BY tmp.c
HAVING count(tmp.ch)=count(DISTINCT tmp.ch)
AND count(tmp.ch)>1
```

2 Найти пары таких покупателей, что список названий товаров, которые они когда-либо покупали, не пересекается ни в одной позиции.

```
SELECT top 3 t1.c, t2.c
FROM
(SELECT soh.CustomerID AS c,
sod.ProductID AS p
FROM [Sales].[SalesORDERDetail] AS sod JOIN
[Sales].[SalesORDERHeader] AS soh
ON sod.SalesORDERID=soh.SalesORDERID) t1,
```

```

(SELECT soh.CustomerID AS c,
sod.ProductID AS p
FROM [Sales].[SalesORDERDetail] AS sod JOIN
[Sales].[SalesORDERHeader] AS soh
ON sod.SalesORDERID=sod.SalesORDERID) t2
WHERE t1.p!=all(SELECT sod.ProductID AS p
FROM [Sales].[SalesORDERDetail] AS sod JOIN
[Sales].[SalesORDERHeader] AS soh
ON sod.SalesORDERID=sod.SalesORDERID
WHERE soh.CustomerID=t2.c)

```

3 Вывести номера продуктов, таких, что их цена выше средней цены продукта в подкатегории, к которой относится продукт. Запрос реализовать двумя способами. В одном из решений допускается использование обобщенного табличного выражения.

Вариант 1

```

SELECT p1.ProductID
FROM [Production].[Product] AS p1
WHERE p1.ListPrice>
(
SELECT avg(p2.[ListPrice])
FROM [Production].[Product] AS p2
WHERE p2.ProductSubcategoryID=p1.ProductSubcategoryID
)

```

Вариант 2

```

with tmp (pscid, acgLP) AS
(SELECT p.ProductSubcategoryID, avg([ListPrice])
FROM [Production].[Product] AS p
GROUP BY p.ProductSubcategoryID)

SELECT p.ProductID
FROM [Production].[Product] AS p JOIN
tmp ON p.ProductSubcategoryID=tmp.pscid
WHERE [ListPrice]>tmp.acgLP

```

Задания для самостоятельной работы

- 1 Найти среднее количество покупок на чек для каждого покупателя (2 способа).
- 2 Найти для каждого продукта и каждого покупателя соотношение количества фактов покупки данного товара данным покупателем к общему количеству фактов покупки товаров данным покупателем
- 3 Вывести на экран следующую информацию: Название продукта, Общее количество фактов покупки этого продукта, Общее количество покупателей этого продукта
- 4 Вывести для каждого покупателя информацию о максимальной и минимальной стоимости одной покупки, чеке, в виде таблицы: номер покупателя, максимальная сумма, минимальная сумма.
- 5 Найти таких номера покупателей, у которых не было нет ни одной пары чеков с одинаковым количеством наименований товаров.
- 6 Найти номера покупателей, у которых все купленные ими товары были куплены как минимум дважды, т.е. на два разных чека.