



SQL

Базовый уровень

Agenda

- Что такое подзапрос
- Подзапросы в
 - WHERE
 - FROM
 - SELECT
 - HAVING
- WITH оператор
- Оператор TOP

Коррелированные и вложенные подзапросы

Коррелированные и вложенные подзапросы

Язык SQL разрешает использовать в других операторах языка DML подзапросы, которые являются внутренними запросами, определяемыми оператором SELECT.

Подзапрос – очень мощное средство языка SQL. Он позволяет строить сложные иерархии запросов, многократно выполняемые в процессе построения результирующего набора или выполнения одного из операторов изменения данных (DELETE, INSERT, UPDATE).

Коррелированные и вложенные подзапросы

Условно подзапросы иногда подразделяют на три типа, каждый из которых является сужением предыдущего:

- табличный подзапрос, возвращающий набор строк и столбцов;
- подзапрос строки, возвращающий только одну строку, но, возможно, несколько столбцов;
- скалярный подзапрос, возвращающий значение одного столбца в одной строке.

Коррелированные и вложенные подзапросы

Подзапрос позволяет решать следующие задачи:

- определять набор строк, добавляемый в таблицу на одно выполнение оператора INSERT;
- определять данные, включаемые в представление, создаваемое оператором CREATE VIEW ;
- определять значения, модифицируемые оператором UPDATE;
- указывать одно или несколько значений во фразах WHERE и HAVING оператора SELECT;

Коррелированные и вложенные подзапросы

Подзапрос позволяет решать следующие задачи:

- определять во фразе FROM таблицу как результат выполнения подзапроса;
- применять коррелированные подзапросы – подзапрос называется коррелированным, если запрос, содержащийся в предикате, имеет ссылку на значение из таблицы (внешней к данному запросу), которая проверяется посредством данного предиката.

Подзапросы в WHERE

WHERE expression [NOT] IN (subquery)

- Сравнение с множеством, формируемым подзапросом

WHERE expression comparison_operator [ANY | ALL] (subquery)

- Сравнение с любым или всеми записями подзапроса

WHERE [NOT] EXISTS (subquery)

- Проверка существования записей в подзапросе

Подзапросы в WHERE

*/*вывести ФИ сотрудников, у которых продажи в прошлом году (поле SalesLastYear) были нулевые*/*

```
select p.FirstName + ' ' + p.LastName as Surname  
from [Person].[Person] p  
where p.BusinessEntityID in (select s.BusinessEntityID  
                             from [Sales].[SalesPerson] s  
                             where s.SalesLastYear = 0)
```

Подзапрос

Подзапросы в WHERE

--Вывести ФИ сотрудниц той же должности, что и сотрудник с логином adventure-works\mark1

```
select p.LastName + ' ' + p.FirstName as Surname
       ,e.JobTitle
from Person.Person p, HumanResources.Employee e
where p.BusinessEntityID = e.BusinessEntityID  --соединение таблиц
       and e.Gender = 'F'                      --сотрудница
       and e.JobTitle in (select JobTitle
                           from HumanResources.Employee
                           where LoginID = 'adventure-works\mark1');
```

Подзапросы в WHERE

--поиск товаров, не принадлежащих категориям велосипедов

```
SELECT Name
FROM Production.Product
WHERE ProductSubcategoryID NOT IN
    (SELECT ProductSubcategoryID
     FROM Production.ProductSubcategory
     WHERE Name = 'Mountain Bikes'
          OR Name = 'Road Bikes'
          OR Name = 'Touring Bikes')
```

Подзапросы в WHERE

--ПОИСК ВСЕХ НАИМЕНОВАНИЙ ТОВАРОВ, ОТНОСЯЩИХСЯ К ПОДКАТЕГОРИИ *Wheels*

SELECT Name

FROM Production.Product **p**

WHERE **EXISTS**

(SELECT *

FROM Production.ProductSubcategory **ps**

WHERE ps.ProductSubcategoryID = p.ProductSubcategoryID

AND Name = 'Wheels')

order by Name

GROUP BY ProductSubcategoryID) ;

Подзапросы в FROM

--число сотрудников в каждом департаменте

-- ВНЕШНЕЕ СОЕДИНЕНИЕ

```
select d.Name, count(ed.DepartmentID) as qty
from HumanResources.Department d left join
    HumanResources.EmployeeDepartmentHistory ed
    on d.DepartmentID = ed.DepartmentID
group by d.DepartmentID, d.Name
order by qty desc
```

--ПОДЗАПРОС В FROM

```
select d.Name, edh.qty
from HumanResources.Department d left join
```

```
(select DepartmentID, count(*) as qty
from HumanResources.EmployeeDepartmentHistory
group by DepartmentID ) edh
```

```
on d.DepartmentID = edh.DepartmentID
order by qty desc
```

Подзапрос

Подзапросы в SELECT

--число сотрудников в каждом департаменте

-- внешнее соединение

```
select d.Name, count(ed.DepartmentID) as qty  
from HumanResources.Department d left join  
    HumanResources.EmployeeDepartmentHistory ed  
    on d.DepartmentID = ed.DepartmentID  
group by d.DepartmentID, d.Name  
order by qty desc
```

-- коррелированный подзапрос

```
select d.Name  
    (select count(*)  
     from HumanResources.EmployeeDepartmentHistory  
     where DepartmentID = d.DepartmentID) as qty  
From HumanResources.Department d  
order by qty desc
```

Подзапрос

Подзапросы в HAVING

--Найти отделы с максимальным числом сотрудников

```
select d.Name, count(ed.DepartmentID) as qty  
from [HumanResources].[EmployeeDepartmentHistory]  
ed,[HumanResources].[Department] d  
where ed.DepartmentID = d.DepartmentID
```

and ed.EndDate is null - ещё работают в этом отделе

```
group by d.DepartmentID, d.Name
```

--размер отдела = найденному максимуму

```
having count(ed.DepartmentID) = (--величина самого большого отдела
```

```
select max(qty)
```

```
from ( --число людей в каждом отделе
```

```
select count(*) as qty
```

```
from [HumanResources].[EmployeeDepartmentHistory]
```

```
where EndDate is null
```

```
group by DepartmentID
```

```
) as DepQty
```

```
)
```


WITH оператор

- Введён в стандарте SQL 99
- Синтаксис:

```
WITH query_name AS (subquery)
    [, query_name AS (subquery) ]...
Select *
From query_name
```

WITH

--Найти отделы с максимальным числом сотрудников

with **t** as

```
(select d.Name, count(*) as qty
from HumanResources.Department d,
     HumanResources.EmployeeDepartmentHistory edh
where edh.DepartmentID = d.DepartmentID
      and edh.EndDate is null -- ещё работают в этом отделе
group by d.DepartmentID, d.Name)
```

select Name, qty

from **t**

where qty = (select max(qty) from **t**)

Подзапрос

Оператор TOP

- Указывает на то, что будет возвращён только первый набор строк из результата запроса.
- Набор строк может быть
 - либо числом,
 - либо процентом.
- Выражение TOP может быть использовано в инструкциях SELECT, INSERT, UPDATE, MERGE и DELETE.
- Синтаксис

TOP (*expression*) [PERCENT]

Оператор TOP

-- вывести 5 сотрудников с максимальными
продажами за прошлый год

```
select TOP (5) BusinessEntityID,SalesLastYear  
from Sales.SalesPerson s  
order by SalesLastYear desc
```



Задаёт список выбора

Оператор TOP

-- вывести 20% сотрудников с максимальными
продажами за прошлый год

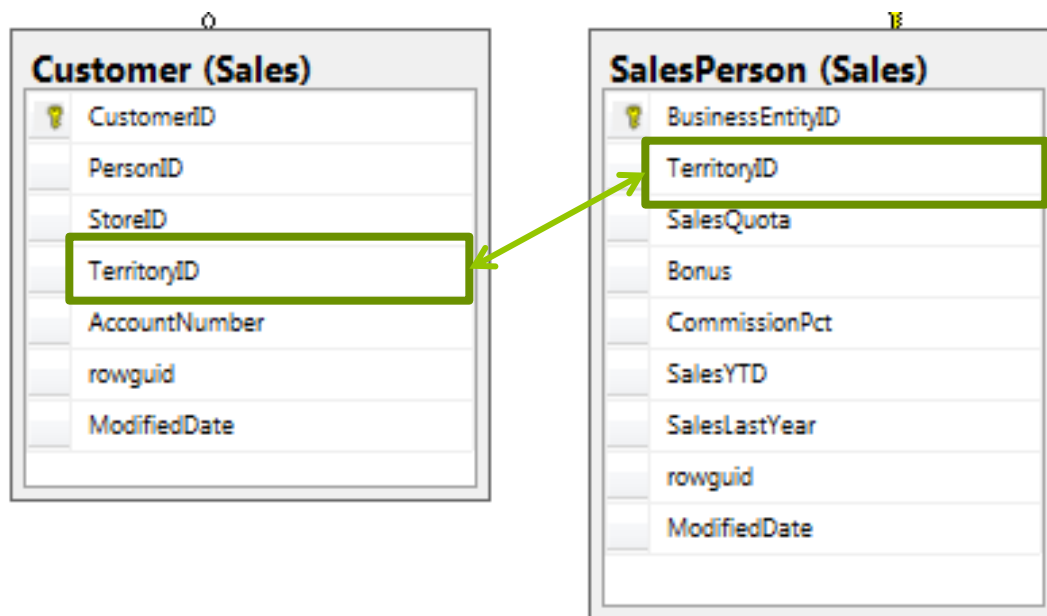
```
select TOP (20) Percent BusinessEntityID,  
           SalesLastYear  
from Sales.SalesPerson s  
order by SalesLastYear desc
```



Задаёт список выбора

Задания

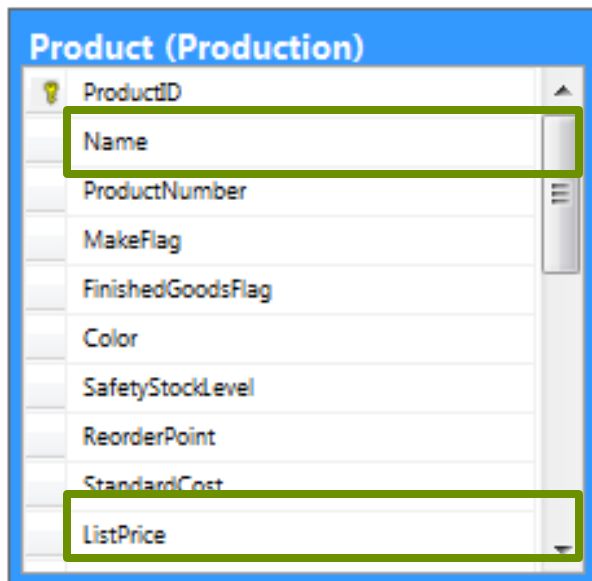
1. Отобразить заказчиков (CustomerID), находящихся на территории, где не работает ни один менеджер по продажам



- с помощью outer join (left или right)
- с помощью подзапроса NOT IN
- с помощью подзапроса ANY

Задания

2. Вывести названия всех продуктов, у которых цена (List Price) по прейскуранту больше, чем средняя цена по прейскуранту



- Отсечение в From
- Отсечение в where

Задание

3. Вывести 5 самых больших отдела с использованием TOP
4. Вывести 5 самых маленьких отдела с учётом TOP

