

## Лекция 9.1

Подзапросы. Виды подзапросов по типу возвращаемого значения. Подзапросы и операторы сравнения, операторы IN, ANY/SOME и ALL. Предикат EXISTS. Подзапросы в инструкции SELECT. Коррелирующие подзапросы.

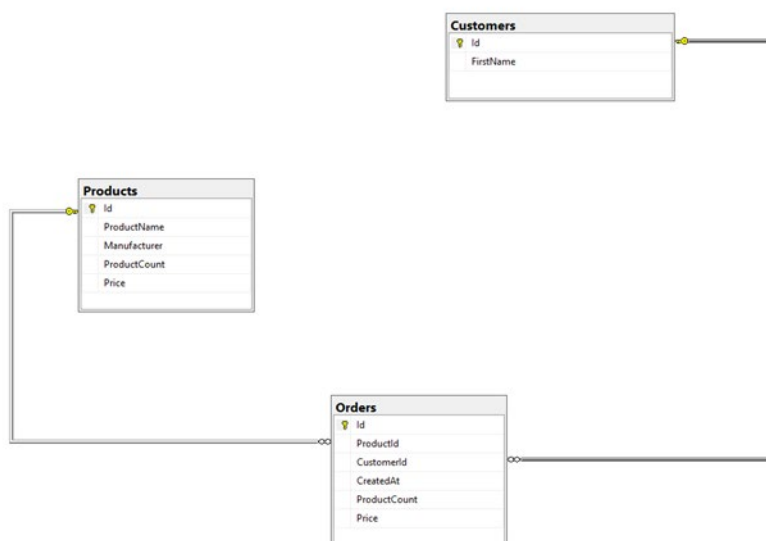
*Наиболее сложной, но в то же время наиболее интересной темой являются подзапросы. Это достаточно мощное средство получения необходимых данных, а с другой стороны, это средство очень сильно бьет по производительности обработки запроса сервером. Сначала мы научимся работать с подзапросами, потому что с их помощью можно быстро решить поставленную задачу, а потом будем учиться избавляться от подзапросов, что идентично оптимизации.*

1. T-SQL поддерживает функциональность подзапросов (subquery), то есть таких запросов, которые могут встроены в другие запросы.

Рассмотрим пример из лекции 7.

```
1 USE testbasa;
2
3 CREATE TABLE Products
4 (
5     Id INT IDENTITY PRIMARY KEY,
6     ProductName NVARCHAR(30) NOT NULL,
7     Manufacturer NVARCHAR(20) NOT NULL,
8     ProductCount INT DEFAULT 0,
9     Price MONEY NOT NULL
10 );
11 CREATE TABLE Customers
12 (
13     Id INT IDENTITY PRIMARY KEY,
14     FirstName NVARCHAR(30) NOT NULL
15 );
16 CREATE TABLE Orders
17 (
18     Id INT IDENTITY PRIMARY KEY,
19     --ON DELETE CASCADE используется для автоматического удаления строк
20     --из дочерней таблицы при удалении строк из родительской таблицы
21     ProductId INT NOT NULL REFERENCES Products(Id) ON DELETE CASCADE,
22     CustomerId INT NOT NULL REFERENCES Customers(Id) ON DELETE CASCADE,
23     CreatedAt DATE NOT NULL,
24     ProductCount INT DEFAULT 1,
25     Price MONEY NOT NULL
26 );
```

Здесь таблицы Products и Customers связаны с таблицей Orders связью один ко многим. Таблица Orders в виде внешних ключей ProductId и CustomerId содержит ссылки на столбцы Id из соответственно таблиц Products и Customers. Также она хранит количество купленного товара (ProductCount) и и по какой цене он был куплен (Price). И кроме того, таблица также хранит в виде столбца CreatedAt дату покупки.



Добавление данных

```

USE testbasa;
INSERT INTO Products
VALUES ('iPhone 6', 'Apple', 2, 36000),
('iPhone 6S', 'Apple', 2, 41000),
('iPhone 7', 'Apple', 5, 52000),
('Galaxy S8', 'Samsung', 2, 46000),
('Galaxy S8 Plus', 'Samsung', 1, 56000),
('Mi 5X', 'Xiaomi', 2, 26000),
('OnePlus 5', 'OnePlus', 6, 38000);

INSERT INTO Customers VALUES ('Tom'), ('Bob'), ('Sam');

INSERT INTO Orders
VALUES
(
....(SELECT Id FROM Products WHERE ProductName= 'Galaxy S8'),
....(SELECT Id FROM Customers WHERE FirstName= 'Tom'),
....'2017-07-11',
....2,
....(SELECT Price FROM Products WHERE ProductName= 'Galaxy S8')
),
(
....(SELECT Id FROM Products WHERE ProductName= 'iPhone 6S'),
....(SELECT Id FROM Customers WHERE FirstName= 'Tom'),
....'2017-07-13',
....1,
....(SELECT Price FROM Products WHERE ProductName= 'iPhone 6S')
),
(
....(SELECT Id FROM Products WHERE ProductName= 'iPhone 6S'),
....(SELECT Id FROM Customers WHERE FirstName= 'Bob'),
....'2017-07-11',
....1,
....(SELECT Price FROM Products WHERE ProductName= 'iPhone 6S')
);

```

! %

Результаты Сообщения

Id	ProductName	Manufacturer	ProductCount	Price
1	iPhone 6	Apple	2	36000,00
2	iPhone 6S	Apple	2	41000,00
3	iPhone 7	Apple	5	52000,00
4	Galaxy S8	Samsung	2	46000,00
5	Galaxy S8 Plus	Samsung	1	56000,00
6	Mi 5X	Xiaomi	2	26000,00
7	OnePlus 5	OnePlus	6	38000,00

## Customers

152 %

Результаты Сообщения

Id	FirstName
1	Tom
2	Bob
3	Sam

## Orders

%

Результаты Сообщения

Id	ProductId	CustomerId	CreatedAt	ProductCount	Price
1	4	1	2017-07-11	2	46000,00
2	2	1	2017-07-13	1	41000,00
3	2	2	2017-07-11	1	41000,00

Здесь интерес представляет добавление элементов в таблицу Orders.

Например, первый заказ был сделан покупателем Tom на товар Galaxy S8.  
Соответственно в таблицу Orders нам надо сохранить информацию о заказе,  
где поле ProductId указывает на Id товара Galaxy S8,  
поле Price - на его цену,  
а поле CustomerId - на Id покупателя Tom.

**Но на момент написания запроса нам может быть неизвестен ни Id покупателя, ни Id товара, ни цена товара. В этом случае можно выполнить подзапрос.**

Подзапрос выполняет команду SELECT и заключается в скобки. В данном же случае при добавлении одного товара выполняется **три подзапроса.**

**Каждый подзапрос возвращает одного скалярное значение, например, числовой идентификатор.**

В данном случае подзапросы выполнялись к другой таблице, но могут выполняться и к той же, к которой вызывается основной запрос.

Например, найдем товары из таблицы Products, которые имеют минимальную цену:

Id	ProductName	Manufacturer	ProductCount	Price
1	iPhone 6	Apple	2	36000,00
2	iPhone 6S	Apple	2	41000,00
3	iPhone 7	Apple	5	52000,00
4	Galaxy S8	Samsung	2	46000,00
5	Galaxy S8 Plus	Samsung	1	56000,00
6	Mi 5X	Xiaomi	2	26000,00
7	OnePlus 5	OnePlus	6	38000,00

```
1 USE testbasa;  
2 --минимальная цена  
3 SELECT *  
4 FROM Products  
5 WHERE Price = (SELECT MIN(Price) FROM Products)
```

Id	ProductName	Manufacturer	ProductCount	Price
6	Mi 5X	Xiaomi	2	26000,00

Или найдем товары, цена которых выше средней:

Id	ProductName	Manufacturer	ProductCount	Price
3	iPhone 7	Apple	5	52000,00
4	Galaxy S8	Samsung	2	46000,00
5	Galaxy S8 Plus	Samsung	1	56000,00

## 2. Коррелирующие подзапросы

Подзапросы бывают коррелирующими и некоррелирующими.

В примерах выше команды SELECT выполняли фактически один подзапрос для всей команды, например, подзапрос возвращает минимальную или среднюю цену, которая не изменится, сколько бы мы строк не выбирали в основном запросе.

*То есть результат подзапроса не зависит от строк, которые выбираются в основном запросе.*

*И такой подзапрос выполняется один раз для всего внешнего запроса.*

Но также существуют коррелирующие подзапросы (correlated subquery), результаты которых зависят от строк, которые выбираются в основном запросе.

### Определения:

**1) Подзапрос**, возвращающий результат или результаты, для получения которых значения указанного столбца не должны соотноситься (коррелировать) со значениями столбцов, указанных в основном запросе, называется некоррелирующим.

*Результат выполнения некоррелирующего запроса не зависит от значений, возвращаемых основным запросом.*

Обычно некоррелирующие запросы применяются в запросах, в которых значение определённого столбца сравнивается со значением, возвращаемым подзапросом, в запросах с предикатом IN, кванторными функциями ALL и ANY.

Однако уже в запросах с предикатом EXISTS применяются коррелирующие подзапросы.

**2) Подзапрос**, возвращающий результат или результаты, для получения которых значения указанного столбца должны соотноситься (коррелировать) со значениями столбцов, указанных в основном запросе, называется коррелирующим.

*Иными словами, результат, выполнения подзапроса зависит от значений, возвращаемых основным запросом.*

Часто коррелирующие подзапросы применяются для получения значений одного из столбцов результирующей таблицы и в этих случаях подзапрос, заключённый в скобки, перечисляется через запятую вместе с именами столбцов из таблиц или соединения таблиц.

*Например, выберем все заказы из таблицы Orders, добавив к ним информацию о товаре:*

Id	ProductId	CustomerId	CreatedAt	ProductCount	Price
1	4	1	2017-07-11	2	46000,00
2	2	1	2017-07-13	1	41000,00
3	2	2	2017-07-11	1	41000,00

Id	ProductName	Manufacturer	ProductCount	Price
1	iPhone 6	Apple	2	36000,00
2	iPhone 6S	Apple	2	41000,00
3	iPhone 7	Apple	5	52000,00
4	Galaxy S8	Samsung	2	46000,00
5	Galaxy S8 Plus	Samsung	1	56000,00
6	Mi 5X	Xiaomi	2	26000,00
7	OnePlus 5	OnePlus	6	38000,00

### Customers

Id	FirstName
1	Tom
2	Bob
3	Sam

*Здесь для каждой строки из таблицы Orders будет выполняться подзапрос, результат которого зависит от столбца ProductId. И каждый подзапрос может возвращать различные данные.*

```

11
12  --выберем все заказы из таблицы Orders, добавив к ним информацию о товаре
13  SELECT CreatedAt,
14          Price,
15          (SELECT ProductName FROM Products
16           WHERE Products.Id = Orders.ProductId) AS Product
17  FROM Orders

```

CreatedAt	Price	Product
2017-07-11	46000,00	Galaxy S8
2017-07-13	41000,00	iPhone 6S
2017-07-11	41000,00	iPhone 6S



**Коррелирующий подзапрос может выполняться и для той же таблицы, к которой выполняется основной запрос.**

**Например, выберем из таблицы *Products* те товары, стоимость которых выше средней цены товаров для данного производителя**

Id	ProductName	Manufacturer	ProductCount	Price
1	iPhone 6	Apple	2	36000,00
2	iPhone 6S	Apple	2	41000,00
3	iPhone 7	Apple	5	52000,00
4	Galaxy S8	Samsung	2	46000,00
5	Galaxy S8 Plus	Samsung	1	56000,00
6	Mi 5X	Xiaomi	2	26000,00
7	OnePlus 5	OnePlus	6	38000,00

```
20 --коррелирующий подзапрос к одной таблице
21 --выберем из таблицы Products те товары,
22 --стоимость которых выше средней цены товаров для данного производителя
23
24 SELECT ProductName,
25         Manufacturer,
26         Price,
27         (SELECT AVG(Price) FROM Products AS SubProds
28          WHERE SubProds.Manufacturer=Prods.Manufacturer) AS AvgPrice
29 FROM Products AS Prods
30 WHERE Price >
31        (SELECT AVG(Price) FROM Products AS SubProds
32         WHERE SubProds.Manufacturer=Prods.Manufacturer)|
33
```

Id	ProductName	Manufacturer	Price	AvgPrice
1	iPhone 7	Apple	52000,00	43000,00
2	Galaxy S8 Plus	Samsung	56000,00	51000,00

В данном случае определено два коррелирующих подзапроса.

Первый подзапрос определяет спецификацию столбца *AvgPrice*. Он будет выполняться для каждой строки, извлекаемой из таблицы *Products*.

В подзапрос передается производитель товара и на его основе выбирается средняя цена для товаров именно этого производителя. И так как производитель у товаров может отличаться, то и результат подзапроса в каждом случае также может отличаться.

Второй подзапрос аналогичен, только он используется для фильтрации извлекаемых из таблицы *Products*. И также он будет выполняться для каждой строки.

**Чтобы избежать двойственности при фильтрации в подзапросе при сравнении производителей (*SubProds.Manufacturer=Prods.Manufacturer*) для внешней выборки установлен псевдоним *Prods*, а для выборки из подзапросов определен псевдоним *SubProds*.**

**!Следует учитывать, что коррелирующие подзапросы выполняются для каждой отдельной строки выборки, то выполнение таких подзапросов может замедлять выполнение всего запроса в целом.**

