

## Лекция 9.2

**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.**

Язык SQL поддерживает написание вложенных запросов. Запрос, который возвращает конечный результат, называют **внешним**. Внутри себя он может использовать **результат** выполнения других запросов.

**Внутренние запросы** вызываются во время выполнения кода и являются аналогами выражений, основанных на переменных или константах. В отличие от выражений их результат может изменяться в зависимости от содержимого таблицы. **Использование вложенных запросов избавляет от необходимости хранить промежуточные результаты в отдельных переменных.**

**Вложенные запросы могут быть либо автономными, либо коррелирующими.** Первые, в отличие от вторых, никак не связаны со своим внешним кодом. **Результат вложенного запроса может состоять из одного или нескольких значений или же вовсе являться таблицей.**

**Подзапросы (вложенные запросы), возвращающие единственное значение**, наиболее часто применяются в случаях, когда значение определённого столбца в основном запросе требуется сравнить с некоторым единственным значением при помощи одного из операторов сравнения (=, <, >, <=, >=). Значение, с которым производится сравнение, как раз и возвращается подзапросом (вложенным запросом).

**Верные признаки того, что подзапрос вернёт одно единственное значение:**

- в подзапросе применяется одна из агрегатных функций (COUNT, SUM, AVG, MAX, MIN);
- подзапрос извлекает значение уникального идентификатора, например, первичного ключа.

**Подзапросы, возвращающие множество значений**

Подзапросы, возвращающие множество значений, могут применяться в запросах с предикатами IN и EXISTS и кванторными функциями ALL и ANY.

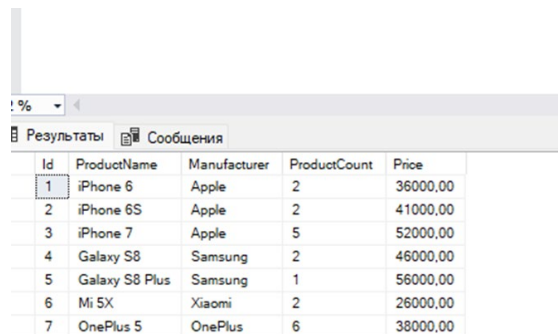
## Подзапросы в основных командах SQL

В выражении **SELECT** мы можем вводить подзапросы четырьмя способами:

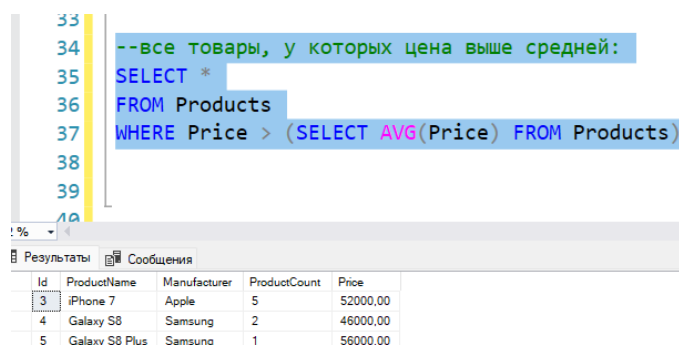
- 1) Использовать в условии в выражении **WHERE**
- 2) Использовать в условии в выражении **HAVING**
- 3) Использовать в качестве таблицы для выборки в выражении **FROM**
- 4) Использовать в качестве спецификации столбца в выражении **SELECT**

### 9.1.1 Рассмотрим некоторые из этих случаев.

Например, получим все товары, у которых цена выше средней:



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



```
33
34 --все товары, у которых цена выше средней:
35 SELECT *
36 FROM Products
37 WHERE Price > (SELECT AVG(Price) FROM Products)
38
39
40
```

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

Чтобы получить нужные товары, нам вначале надо выполнить подзапрос на получение средней цены товара: `SELECT AVG(Price) FROM Products`.

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

### Orders

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

Или выберем всех покупателей из таблицы Customers, у которых нет заказов в таблице Orders:

```

38
39 --выберем всех покупателей из таблицы Customers, у которых нет заказов в таблице Orders
40
41 SELECT * FROM CUSTOMERS
42 WHERE Id NOT IN (SELECT CustomerId FROM Orders)
43
44
45

```

Id	FirstName
3	Sam

Хотя в данном случае подзапросы прекрасно справляются со своей задачей, стоит отметить, что это не самый эффективный способ для извлечения данных из других таблиц, так как в рамках T-SQL для сведения данных из разных таблиц **можно использовать оператор JOIN**.

### 9.2.2 Получение набора значений

При использовании в операторах сравнения подзапросы должны возвращать одно скалярное значение.

Скалярный вложенный запрос всегда возвращает одно значение независимо от того, автономный он или коррелирующий. Он может находиться в любом месте внешнего кода, где допустимо использование скалярных выражений (например, в инструкциях WHERE или SELECT).

Но иногда возникает необходимость получить набор значений. Чтобы при использовании в операторах сравнения подзапрос мог возвращать набор значений, перед ним необходимо использовать один из операторов: ALL, SOME или ANY.

При использовании **ключевого слова ALL** условие в операции сравнения должно быть верно для всех значений, которые возвращаются подзапросом.

Например, найдем все товары, цена которых меньше чем у любого товара фирмы Apple

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

```
43
44 --найдем все товары, цена которых меньше чем у любого товара фирмы Apple
45 SELECT * FROM Products
46 WHERE Price < ALL(SELECT Price FROM Products WHERE Manufacturer='Apple')
47
```

%

РезультатыСообщения

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

Если бы мы в данном случае опустили бы ключевое слово **ALL**, то мы бы столкнулись с ошибкой

```
47
48 --без ALL
49 SELECT * FROM Products
50 WHERE Price < (SELECT Price FROM Products WHERE Manufacturer='Apple')
51
```

Сообщение 512, уровень 16, состояние 1, строка 49  
Subquery returned more than 1 value. This is not permitted when the subquery follows =, !=, <, <=, >, >= or when the subquery is used as a

Время выполнения: 2023-04-06T16:02:59.4218369+03:00

В тоже время подобный запрос гораздо проще переписать другим образом:

```
50 --если подзапрос возвращает значения
51 SELECT * FROM Products
52 WHERE Price < (SELECT MIN(Price) FROM Products WHERE Manufacturer='Apple')
```

152 %													
Результаты	Сообщения												
<table> <tr> <th></th><th>Id</th><th>ProductName</th><th>Manufacturer</th><th>ProductCount</th><th>Price</th></tr> <tr> <td>1</td><td>6</td><td>Mi 5X</td><td>Xiaomi</td><td>2</td><td>26000,00</td></tr> </table>		Id	ProductName	Manufacturer	ProductCount	Price	1	6	Mi 5X	Xiaomi	2	26000,00	
	Id	ProductName	Manufacturer	ProductCount	Price								
1	6	Mi 5X	Xiaomi	2	26000,00								

*При применении ключевых слов **ANY** и **SOME** условие в операции сравнения должно быть истинным для хотя бы одного из значений, возвращаемых подзапросом.*

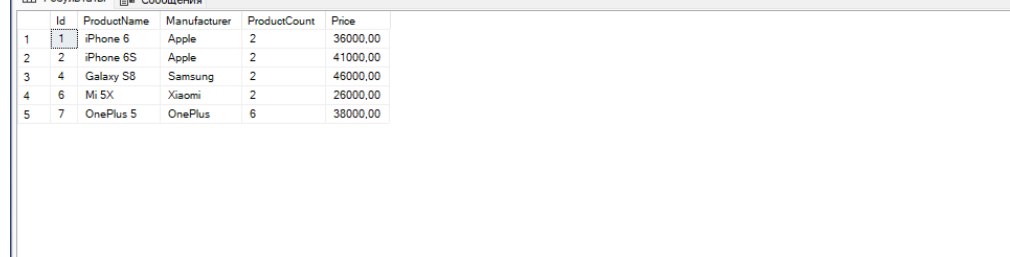
*По действию оба этих оператора аналогичны, поэтому можно применять любое из них.*

*Например, в следующем случае получим товары, которые стоят меньше самого дорогого товара компании Apple:*



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


```
54
55 --товары, которые стоят меньше самого дорогого товара компании Apple
56
57 SELECT * FROM Products
58 WHERE Price < ANY(SELECT Price FROM Products WHERE Manufacturer='Apple')
```



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

*И также стоит отметить, что данный запрос можно сделать проще, переписав следующим образом:*

```
59
60 SELECT * FROM Products
61 WHERE Price < (SELECT MAX(Price) FROM Products WHERE Manufacturer='Apple')
```



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

### 9.2.3 Подзапрос как спецификация столбца

Результат подзапроса может представлять отдельный столбец в выборке.

Например, выберем все заказы и добавим к ним информацию о названии товара

	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

```
62
63 --все заказы и добавим к ним информацию о названии товара
64
65 SELECT *,
66 (SELECT ProductName FROM Products WHERE Id=Orders.ProductId) AS Product
67 FROM Orders
```

152 %							
<div>РезультатыСообщения</div>							
	Id	ProductId	CustomerId	CreatedAt	ProductCount	Price	Product
1	4	1	2017-07-11	2	46000,00	Galaxy S8	
2	2	1	2017-07-13	1	41000,00	iPhone 6S	
3	2	2	2017-07-11	1	41000,00	iPhone 6S	



### 9.2.3 Предикат EXISTS

Язык T-SQL поддерживает предикат под названием **EXISTS**, который возвращает **TRUE** или **FALSE** в зависимости от того, генерирует ли переданный ему вложенный запрос какие-либо строки.

Предикат EXISTS хорошо поддается **оптимизации**. Ядро Microsoft SQL Server знает, **что не нужно обрабатывать все запрашиваемые строки** - достаточно получить хотя бы один результат (или его отсутствие).

**Это тот редкий случай, когда в использовании символа звездочки (\*) внутри инструкции SELECT нет ничего плохого.**

Предикат **EXISTS** следит исключительно за наличием **подходящих строк, игнорируя атрибуты**, указанные для выборки.

Ядро SQL Server знает об этом и в целях оптимизации пропускает обработку списка **SELECT** внутри вложенного запроса. Поэтому использование группового символа \* вместо перечисления констант не влияет на производительность. Справедливости ради нужно отметить, что на процесс поиска столбцов все же могут быть выделены незначительные ресурсы.

Следует упомянуть об еще одной интересной особенности инструкции **EXISTS**: в отличие от большинства других предикатов в языке T-SQL она использует **двоичную логику**. Действительно, не существует такой ситуации, в которой на вопрос «возвращает ли код строки?» нельзя было бы ответить однозначно.

**Оператор EXISTS позволяет проверить, возвращает ли подзапрос какое-либо значение.**

**Как правило, этот оператор используется для индикации того, что какая-либо строка удовлетворяет условию.**

**То есть фактически оператор EXISTS не возвращает строки, а лишь указывает, что в базе данных есть как минимум одна строка, которые соответствует данному запросу.**

**Поскольку возвращения набора строк не происходит, то подзапросы с подобным оператором выполняются довольно быстро.**

**Применение оператора имеет следующий формальный синтаксис:**

***WHERE [NOT] EXISTS (подзапрос)***

**Например, найдем всех покупателей из таблицы Customer, которые делали заказы**

The screenshot shows a database query interface with three tables: Customers, Orders, and a query result. The Customers table has columns Id, FirstName, and LastName. The Orders table has columns Id, ProductId, CustomerId, CreatedAt, ProductCount, and Price. The query result shows the output of the SQL query: `SELECT * FROM Customers WHERE EXISTS (SELECT * FROM Orders WHERE Orders.CustomerId = Customers.Id)`. The result shows two customers: Tom (Id 1) and Bob (Id 2).

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

Id	FirstName
1	Tom
2	Bob
3	Sam

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

```
69 --найдем всех покупателей из таблицы Customer, которые делали заказы
70
71 SELECT *
72 FROM Customers
73 WHERE EXISTS (SELECT * FROM Orders
74               WHERE Orders.CustomerId = Customers.Id)
```

Id	FirstName
1	Tom
2	Bob

**Другой пример - найдем все товары из таблицы Products, на которые не было заказов в таблице Orders:**

The screenshot shows a database query interface with the Products table and a query result. The Products table has columns Id, ProductName, Manufacturer, ProductCount, and Price. The query result shows the output of the SQL query: `SELECT * FROM Products WHERE NOT EXISTS (SELECT * FROM Orders WHERE Products.Id = Orders.ProductId)`. The result shows five products: iPhone 6 (Id 1), iPhone 7 (Id 3), Galaxy S8 Plus (Id 5), Mi 5X (Id 6), and OnePlus 5 (Id 7).

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

```
75
76 --найдем все товары из таблицы Products,
77 --на которые не было заказов в таблице Orders
78 SELECT *
79 FROM Products
80 WHERE NOT EXISTS (SELECT * FROM Orders WHERE Products.Id = Orders.ProductId)
```

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

Подзапрос используется для возврата данных, которые будут использоваться в основном запросе, **в качестве условия для дальнейшей фильтрации данных, подлежащих извлечению.**

Подзапросы могут использоваться с инструкциями **SELECT, INSERT, UPDATE и DELETE** вместе с операторами типа =, <, >, >=, <=, IN, BETWEEN и т. д.

**Существует несколько правил, которые применяются к подзапросам:**

*Подзапросы должны быть заключены в круглые скобки.*

*Подзапрос может иметь только один столбец в условии SELECT, если только несколько столбцов не указаны в основном запросе для подзапроса для сравнения выбранных столбцов.*

*Команда ORDER BY не может использоваться в подзапросе, хотя в основном запросе она использоваться может. В подзапросе может использоваться команда GROUP BY для выполнения той же функции, что и ORDER BY.*

*Подзапросы, которые возвращают более одной строки, могут использоваться только с несколькими операторами значений, такими как оператор IN.*

*Подзапрос не может быть сразу заключен **в функцию set.***

*С подзапросом не может использоваться оператор BETWEEN. Однако оператор BETWEEN может использоваться внутри подзапроса.*