

# Sub Query in MS-SQL Server

---



Soyeb Ghachi

Technical Trainer | Mentor | SME

Sub Query

# Overview



## SUB QUERY

- In place of an expression
- With IN or NOT IN
- With ANY or ALL
- With EXISTS or NOT EXISTS
- All

# SUB-QUERY

---

## Subquery

A sub query is a query nested inside another statement such as SELECT, INSERT, UPDATE, or DELETE. Let's see the following example.

The following statement shows how to use a subquery in the WHERE clause of a SELECT statement to find the sales orders of the customers who locate in New York:

```
SELECT column1, column2, ...  
FROM (SELECT column1, column2, ...  
      FROM subquery_table  
      WHERE condition) AS derived_table;
```

```
SELECT
    order_id,
    order_date,
    customer_id
FROM
    orders
WHERE
    customer_id IN (
        SELECT
            customer_id
        FROM
            customers
        WHERE
            city = 'New York'
    )
ORDER BY
    order_date DESC;
```

outer query

subquery

Sub Query

## **SQL Server subquery types:**

- In place of an expression
- With IN or NOT IN
- With ANY or ALL
- With EXISTS or NOT EXISTS
- All

## A.SQL Server subquery is used in place of an expression

If a subquery returns a single value, it can be used anywhere an expression is used.

```
SELECT
    order_id,
    order_date,
    (
        SELECT
            MAX (list_price)
        FROM
            order_items i
        WHERE
            i.order_id = o.order_id
    ) AS max_list_price
FROM
    orders o
order by order_date desc;
```

order_id	order_date	max_list_price
1615	2018-12-28	2499.99
1614	2018-11-28	2299.99
1613	2018-11-18	4999.99
1612	2018-10-21	1559.99
1611	2018-09-06	3199.99
1610	2018-08-25	3199.99
1609	2018-08-23	749.99
1608	2018-07-12	529.99
1607	2018-07-11	999.99
1606	2018-07-10	659.99
1605	2018-07-01	4499.99
1604	2018-06-17	209.99
1602	2018-04-30	899.99
1603	2018-04-30	229.99
1598	2018-04-29	3499.99
1599	2018-04-29	481.99

**B. SQL Server subquery is used with IN operator.** A subquery which is used with the IN operator returns a set of zero or more values. After the subquery returns values, the outer query makes use of them.

```
SELECT
    product_id,
    product_name
FROM
    products
WHERE
    category_id IN (
        SELECT
            category_id
        FROM
            categories
        WHERE
            category_name = 'Mountain Bikes'
            OR category_name = 'Road Bikes'
    );
```

product_id	product_name
1	Trek 820 - 2016
2	Ritchey Timberwolf Frameset - 2016
3	Surly Wednesday Frameset - 2016
4	Trek Fuel EX 8 29 - 2016
5	Heller Shagamaw Frame - 2016
6	Surly Ice Cream Truck Frameset - 2016
7	Trek Slash 8 27.5 - 2016
8	Trek Remedy 29 Carbon Frameset - 2016
27	Surly Big Dummy Frameset - 2017
28	Surly Karate Monkey 27.5+ Frameset - 2017
29	Trek X-Caliber 8 - 2017
30	Surly Ice Cream Truck Frameset - 2017
31	Surly Wednesday - 2017



**C. SQL Server subquery is used with ANY operator.** The following query finds the products whose list price is greater than or equal to the maximum list price returned by the subquery:

```
SELECT
    product_name,
    list_price
FROM
    products
WHERE
    list_price >= ANY (
        SELECT
            AVG (list_price)
        FROM
            products
        GROUP BY
            brand_id
    )
```

product_name	list_price
Trek 820 - 2016	379.99
Ritchey Timberwolf Frameset - 2016	749.99
Surly Wednesday Frameset - 2016	999.99
Trek Fuel EX 8 29 - 2016	2899.99
Heller Shagawaw Frame - 2016	1320.99
Surly Ice Cream Truck Frameset - 2016	469.99
Trek Slash 8 27.5 - 2016	3999.99
Trek Remedy 29 Carbon Frameset - 2016	1799.99
Trek Conduit+ - 2016	2999.99
Surly Straggler - 2016	1549.00
Surly Straggler 650b - 2016	1680.99
Electra Townie Original 21D - 2016	549.99
Electra Cruiser 1 (24-Inch) - 2016	269.99
Electra Girl's Hawaii 1 (16-inch) - 2015/2016	269.99
Electra Moto 1 - 2016	529.99

**D. ALL Operator:** The ALL operator returns TRUE if all comparison pairs (scalar\_expression, vi) evaluate to TRUE; otherwise, it returns FALSE.

```
SELECT
    product_name,
    list_price
FROM
    products
WHERE
    list_price >= ALL (
        SELECT
            AVG (list_price)
        FROM
            products
        GROUP BY
            brand_id
    )
```

product_name	list_price
Trek Fuel EX 8 29 - 2016	2899.99
Trek Slash 8 27.5 - 2016	3999.99
Trek Conduit+ - 2016	2999.99
Trek Fuel EX 9.8 29 - 2017	4999.99
Trek Fuel EX 9.8 27.5 Plus - 2017	5299.99
Trek Remedy 9.8 - 2017	5299.99
Trek Domane SL 6 - 2017	3499.99
Trek Silque SLR 7 Women's - 2017	5999.99
Trek Silque SLR 8 Women's - 2017	6499.99
Trek Domane SL Disc Frameset - 2017	3199.99
Trek Domane S 6 - 2017	2699.99
Trek Domane SLR 6 Disc - 2017	5499.99
Trek Madone 9.2 - 2017	4999.99
Trek Domane S 5 Disc - 2017	2599.99
Trek Powerfly 8 FS Plus - 2017	4999.99

## E. SQL Server subquery is used with EXISTS or NOT EXISTS.

The EXISTS operator returns TRUE if the subquery return results; otherwise it returns FALSE. On the other hand, the NOT EXISTS is opposite to the EXISTS operator.

```
SELECT
    customer_id,
    first_name,
    last_name,
    city
FROM
    customers c
WHERE
    EXISTS (
        SELECT
            customer_id
        FROM
            orders o
        WHERE
            o.customer_id = c.customer_id
            AND YEAR (order_date) = 2017
    )
ORDER BY
    first_name,
    last_name;
```

customer_id	first_name	last_name	city
75	Abby	Gamble	Amityville
1224	Abram	Copeland	Harlingen
673	Adam	Henderson	Los Banos
1023	Adena	Blake	Ballston Spa
1412	Adrien	Hunter	Rego Park
769	Agatha	Melton	Springfield Gardens
771	Agnes	Sims	Buffalo
1181	Agustina	Lawrence	Brooklyn
735	Aide	Franco	Atwater
384	Aimee	Meritt	Flushing
1093	Alejandrina	Hodges	Deer Park
534	Alejandro	Haney	Wantagh
562	Alejandro	Norman	Upland

Sub Query

## E. SQL Server subquery is used with NOT EXISTS.

```
SELECT
    customer_id,
    first_name,
    last_name,
    city
FROM
    customers c
WHERE
    NOT EXISTS (
        SELECT
            customer_id
        FROM
            orders o
        WHERE
            o.customer_id = c.customer_id
            AND YEAR (order_date) = 2017
    )
ORDER BY
    first_name,
    last_name;
```

customer_id	first_name	last_name	city
1174	Aaron	Knapp	Yonkers
338	Abbey	Pugh	Forest Hills
1085	Adam	Thomton	Central Islip
195	Addie	Hahn	Franklin Square
1261	Adelaida	Hancock	San Pablo
22	Adelle	Larsen	East Northport
574	Adriene	Rivera	Encino
1252	Adriene	Rollins	Plainview
527	Afton	Juarez	Coram
1353	Agatha	Daniels	South El Monte
1322	Ai	Forbes	Franklin Square
937	Aida	Koch	West Hempstead

## References

<https://learn.microsoft.com/en-us/sql/relational-databases/performance/joins>

[https://www.w3schools.com/sql/sql\\_where.asp](https://www.w3schools.com/sql/sql_where.asp)

<https://www.sqlservertutorial.net/sql-server-basics/sql-server-where/>

<https://www.javatpoint.com/sql-server-joins>

<https://www.javatpoint.com/sql-server-joins>

<https://www.guru99.com/sql-server-joins.html>

<https://learn.microsoft.com/en-us/sql/relational-databases/performance/subqueries>