



SQL Server: Subqueries

This SQL Server tutorial explains how to use **subqueries** in SQL Server (Transact-SQL) with syntax and examples.

What is a subquery in SQL Server?

In SQL Server, a subquery is a query within a query. You can create subqueries within your SQL statements. These subqueries can reside in the WHERE clause, the FROM clause, or the SELECT clause.

Note

- In SQL Server (Transact-SQL), a subquery is also called an INNER QUERY or INNER SELECT.
- In SQL Server (Transact-SQL), the main query that contains the subquery is also called the OUTER QUERY or OUTER SELECT.

WHERE clause

Most often, the subquery will be found in the WHERE clause. These subqueries are also called **nested subqueries**.

For example:

```
SELECT p.product_id, p.product_name
FROM products p
WHERE p.product_id IN
    (SELECT inv.product_id
     FROM inventory inv
     WHERE inv.quantity > 10);
```

The subquery portion of the SELECT statement above is:

```
(SELECT inv.product_id
 FROM inventory inv
 WHERE inv.quantity > 10);
```

This subquery allows you to find all *product_id* values from the inventory table that have a quantity greater than 10. The subquery is then used to filter the results from the main query using the IN condition.

This subquery could have alternatively been written as an INNER join as follows:

```
SELECT p.product_id, p.product_name
FROM products p
INNER JOIN inventory inv
ON p.product_id = inv.product_id
WHERE inv.quantity > 10;
```

This INNER JOIN would run more efficiently than the original subquery. It is important to note, though, that not all subqueries can be rewritten using joins.

FROM clause

A subquery can also be found in the FROM clause. These are called **inline views**.

For example:

```
SELECT suppliers.supplier_name, subquery1.total_amt
FROM suppliers,
  (SELECT supplier_id, SUM(orders.amount) AS total_amt
   FROM orders
   GROUP BY supplier_id) subquery1
WHERE subquery1.supplier_id = suppliers.supplier_id;
```

In this example, we've created a subquery in the FROM clause as follows:

```
(SELECT supplier_id, SUM(orders.amount) AS total_amt
 FROM orders
 GROUP BY supplier_id) subquery1
```

This subquery has been aliased with the name *subquery1*. This will be the name used to reference this subquery or any of its fields.

SELECT clause

A subquery can also be found in the SELECT clause. These are generally used when you wish to retrieve a calculation using an aggregate function such as the SUM, COUNT, MIN, or MAX function, but you do not want the aggregate function to apply to the main query.

For example:

```
SELECT e1.last_name, e1.first_name,  
       (SELECT MAX(salary)  
        FROM employees e2  
        WHERE e1.employee_id = e2.employee_id) subquery2  
FROM employees e1;
```

In this example, we've created a subquery in the SELECT clause as follows:

```
(SELECT MAX(salary)  
 FROM employees e2  
 WHERE e1.employee_id = e2.employee_id) subquery2
```

The subquery has been aliased with the name *subquery2*. This will be the name used to reference this subquery or any of its fields.

The trick to placing a subquery in the select clause is that the subquery must return a single value. This is why an aggregate function such as the SUM, COUNT, MIN, or MAX function is commonly used in the subquery.

Copyright TechOnTheNet.com