# **SQL Correlated Subquery**

## **Introduction to SQL correlated subquery**

A correlated subquery is a subquery that depends on the outer query. It means that the[WHERE clause](http://www.zentut.com/sql-tutorial/sql-where/) of the correlated subquery uses the data of the outer query.

The main difference between a correlated subquery and a non-correlated subquery is that you cannot execute a correlated subquery alone like a non-correlated subquery. In addition, a correlated subquery executes once for each selected row from the outer query.

A correlated subquery is also known as repeating subquery or synchronized subquery.

## **SQL correlated subquery examples**

Let’s take a look at some examples to understand the idea of the correlated subquery.

### SQL correlated subquery in the SELECT clause example

The following query selects top five customers by sales:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | SELECT      companyname,      city,      (SELECT              SUM(unitprice \* quantity)       FROM              orders       INNER JOIN              orderdetails ON orderdetails.orderid = orders.orderid       WHERE              orders.customerid = customers.customerid) AS total  FROM      customers  ORDER BY total DESC  LIMIT 5; |

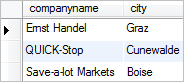


The correlated subquery calculates total sales for each selected customer from thecustomers table. The selected customerid from the outer query is passed to the correlated subquery for getting the corresponding sales data.

### SQL correlated subquery in WHERE clause example

You can also use the correlated subquery in a [WHERE clause](http://www.zentut.com/sql-tutorial/sql-where/). For example, the following example uses a correlated subquery in WHERE clause to find customers that have total sales more than 100K:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | SELECT      companyname, city  FROM      customers  WHERE      100000 < (          SELECT              SUM(unitprice \* quantity)          FROM              orders          INNER JOIN              orderdetails ON orderdetails.orderid = orders.orderid          WHERE              orders.customerid = customers.customerid); |

[](http://www.zentut.com/wp-content/uploads/2012/10/sql-correlated-subquery-in-where-clause-example.png)

For each customer, the correlated subquery calculates the total sales. The WHERE clause checks if the total sales, which is returned by the correlated subquery, is greater than 100K.

### SQL correlated subquery in HAVING clause example

You can use a correlated subquery in the [HAVING clause](http://www.zentut.com/sql-tutorial/sql-having/) of an outer query. See the following example:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | SELECT      t1.categoryID, categoryName  FROM      products t1  INNER JOIN      categories c ON c.categoryID = t1.categoryID  GROUP BY categoryID  HAVING MAX(unitprice) > ALL (     SELECT  2 \* AVG(unitprice)     FROM          products t2      WHERE          t1.categoryID = t2.categoryID) |



In the above query:

* The subquery calculate the average unit price in each category and multiply it with 2.
* The outer query selects category of the product whose unit price is greater than the double average unit price returned by the correlated subquery.