Introduction to SQL Server GROUP BY clause

The GROUP BY clause allows you to arrange the rows of a [query](https://www.sqlservertutorial.net/sql-server-basics/sql-server-select/) in groups. The groups are determined by the columns that you specify in the GROUP BY clause.

The following illustrates the GROUP BY clause syntax:

SELECT

select\_list

FROM

table\_name

GROUP BY

column\_name1,

column\_name2 ,...;

Code language: SQL (Structured Query Language) (sql)

In this query, the GROUP BY clause produced a group for each combination of the values in the columns listed in the GROUP BY clause.

Consider the following example:

SELECT

customer\_id,

YEAR (order\_date) order\_year

FROM

sales.orders

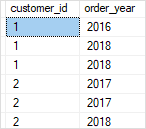
WHERE

customer\_id IN (1, 2)

ORDER BY

customer\_id;

Code language: SQL (Structured Query Language) (sql)



In this example, we retrieved the customer id and the ordered year of the customers with customer id one and two.

As you can see clearly from the output, the customer with the id one placed one order in 2016 and two orders in 2018. The customer with id two placed two orders in 2017 and one order in 2018.

Let’s add a GROUP BY clause to the query to see the effect:

SELECT

customer\_id,

YEAR (order\_date) order\_year

FROM

sales.orders

WHERE

customer\_id IN (1, 2)

GROUP BY

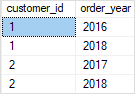
customer\_id,

YEAR (order\_date)

ORDER BY

customer\_id;

Code language: SQL (Structured Query Language) (sql)



The GROUP BY clause arranged the first three rows into two groups and the next three rows into the other two groups with the unique combinations of the customer id and order year.

Functionally speaking, the GROUP BY clause in the above query produced the same result as the following query that uses the [DISTINCT](https://www.sqlservertutorial.net/sql-server-basics/sql-server-select-distinct/) clause:

SELECT DISTINCT

customer\_id,

YEAR (order\_date) order\_year

FROM

sales.orders

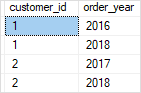
WHERE

customer\_id IN (1, 2)

ORDER BY

customer\_id;

Code language: SQL (Structured Query Language) (sql)



SQL Server GROUP BY clause and aggregate functions

In practice, the GROUP BY clause is often used with [aggregate functions](https://www.sqlservertutorial.net/sql-server-aggregate-functions/) for generating summary reports.

An [aggregate function](https://www.sqlservertutorial.net/sql-server-aggregate-functions/) performs a calculation on a group and returns a unique value per group. For example, [COUNT()](https://www.sqlservertutorial.net/sql-server-aggregate-functions/sql-server-count/) returns the number of rows in each group. Other commonly used aggregate functions are [SUM()](https://www.sqlservertutorial.net/sql-server-aggregate-functions/sql-server-sum/), [AVG()](https://www.sqlservertutorial.net/sql-server-aggregate-functions/sql-server-avg/) (average), [MIN()](https://www.sqlservertutorial.net/sql-server-aggregate-functions/sql-server-min/) (minimum), [MAX()](https://www.sqlservertutorial.net/sql-server-aggregate-functions/sql-server-max/) (maximum).

The GROUP BY clause arranges rows into groups and an aggregate function returns the summary (count, min, max, average, sum, etc.,) for each group.

For example, the following query returns the number of orders placed by the customer by year:

SELECT

customer\_id,

YEAR (order\_date) order\_year,

COUNT (order\_id) order\_placed

FROM

sales.orders

WHERE

customer\_id IN (1, 2)

GROUP BY

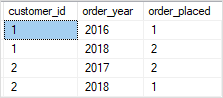
customer\_id,

YEAR (order\_date)

ORDER BY

customer\_id;

Code language: SQL (Structured Query Language) (sql)



If you want to reference a column or expression that is not listed in the GROUP BY clause, you must use that column as the input of an [aggregate function](https://www.sqlservertutorial.net/sql-server-aggregate-functions/). Otherwise, you will get an error because there is no guarantee that the column or expression will return a single value per group. For example, the following query will fail:

SELECT

customer\_id,

YEAR (order\_date) order\_year,

order\_status

FROM

sales.orders

WHERE

customer\_id IN (1, 2)

GROUP BY

customer\_id,

YEAR (order\_date)

ORDER BY

customer\_id;

Code language: SQL (Structured Query Language) (sql)

More GROUP BY clause examples

Let’s take some more examples to understand how the GROUP BY clause works.

Using GROUP BY clause with the COUNT() function example

The following query returns the number of customers in every city:

SELECT

city,

COUNT (customer\_id) customer\_count

FROM

sales.customers

GROUP BY

city

ORDER BY

city;

Code language: SQL (Structured Query Language) (sql)



In this example, the GROUP BY clause groups the customers together by city and the COUNT() function returns the number of customers in each city.

Similarly, the following query returns the number of customers by state and city.

SELECT

city,

state,

COUNT (customer\_id) customer\_count

FROM

sales.customers

GROUP BY

state,

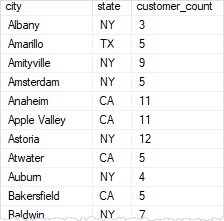
city

ORDER BY

city,

state;

Code language: SQL (Structured Query Language) (sql)



Using GROUP BY clause with the MIN and MAX functions example

The following statement returns the minimum and maximum list prices of all products with the model 2018 by brand:

SELECT

brand\_name,

MIN (list\_price) min\_price,

MAX (list\_price) max\_price

FROM

production.products p

INNER JOIN production.brands b ON b.brand\_id = p.brand\_id

WHERE

model\_year = 2018

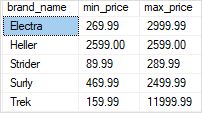
GROUP BY

brand\_name

ORDER BY

brand\_name;

Code language: SQL (Structured Query Language) (sql)



In this example, the WHERE clause is processed before the GROUP BY clause, as always.

Using GROUP BY clause with the AVG() function example

The following statement uses the AVG() function to return the average list price by brand for all products with the model year 2018:

SELECT

brand\_name,

AVG (list\_price) avg\_price

FROM

production.products p

INNER JOIN production.brands b ON b.brand\_id = p.brand\_id

WHERE

model\_year = 2018

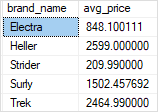
GROUP BY

brand\_name

ORDER BY

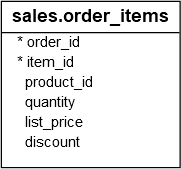
brand\_name;

Code language: SQL (Structured Query Language) (sql)



Using GROUP BY clause with SUM function example

See the following order\_items table:



The following query uses the SUM() function to get the net value of every order:

SELECT

order\_id,

SUM (

quantity \* list\_price \* (1 - discount)

) net\_value

FROM

sales.order\_items

GROUP BY

order\_id;