# Brief definition of grouping in SQL

Grouping in SQL allows us to perform aggregate functions (most of the time) on a set of data that are grouped by a particular criterion (or column).

# Aggregate functions

Commonly used aggregate functions include AVG() or average, COUNT(), MIN(), MAX() and SUM(). Aggregate functions take a set of data, performs calculations on them, and returns a single value derived from that data set. For example, AVG() is going to find the average of a set of data. If you run AVG() on a particular column, you will get the average of all the values on that column. For example, you may want to find the average skill level of employees over the life of the company. You would run something like:

SELECT AVG(skill\_level)

FROM employees; #employees is the table name

The output will be a single column named AVG(skill\_level) with a single row containing the average skill level of all employees throughout the life of the company.

| **AVG(skill\_level)** |
| --- |
| 18.714285714285715 |

# Why group?

Now imagine you want to find the average skill level of employees per year, not the average skill level of employees from the beginning of the company till now. This is where grouping comes in. You can group the average skill level by year so that the output will display two columns, one column named year and another named AVG(skill\_level). Each row will display a year and the year’s average employee skill level. Referring back to the definition of grouping, AVG() is the aggregate function, skill\_level is the set of data and the criterion (column) used to group the data is year. In other words, the below block of code will output the average skill level per year.

SELECT year, AVG(skill\_level)

FROM employees; --employees is the table name

GROUP BY year; --calculate the average per year or group the

--average by year

| **year** | **AVG(skill\_level)** |
| --- | --- |
| 2010 | 10 |
| 2011 | 10 |
| 2012 | 60 |
| 2013 | 21 |