SQL Arithmetic Operator Examples

In this reading, you'll learn more about the arithmetic operators that can be used with SQL. You've learned about the arithmetic operators in SQL that are used to perform basic mathematical operations such as addition, subtraction, multiplication and division. You've also explored the modulus operator, which gives the remainder of a mathematical division. The main objective of this reading is to present some more examples of how arithmetic operators can be used. It also includes more advanced scenarios.

Arithmetic operators

Arithmetic operators are useful when you want to perform mathematical operations on the data in tables while you retrieve them by writing SQL SELECT queries. In SQL, arithmetic operators are used to perform mathematical operations on data. To be more specific, they're used with numerical data stored in database tables.
Arithmetic operators can be used in the SELECT clause as well as in the WHERE clause in a SQL SELECT statement. When an operator is used in the WHERE clause, it's intended to perform the operations on specific rows only. This is because the WHERE clause in SQL is used to filter out data that a particular SQL statement is working on.
All arithmetic operators are used on numerical operands for performing:

Addition

Subtraction

Multiplication

Division

Modulus

Using the addition operator

The SQL addition operator performs the mathematical addition operation on numerical data within columns in a table. For example, if you want to add the values of two instances of numerical data from two separate columns in the table, then you need to specify the two columns as the first and second operand. The syntax is as follows:

```sql
SELECT column_name1 + column_name2 FROM table_name;
```

Let's review an example. This is an employee table from a company database.

employee_ID	**employee_name**	**salary**	**allowance**
1	Alex	25000	1000
2	John	55000	1000
3	James	52000	1000
4	Sam	30000	1000

If you want to know the total salaries of all employees with the basic salary and the allowance added to it then you can use the addition operator. The SQL syntax for the addition operator is as follows:

```sql
SELECT salary + allowance FROM employee;
```

Here, the salary column and the allowance column are the two operands. The addition operator is used to add the values of these twocolumns together. The output is as follows:

Salary + allowance
26000
56000
53000
31000

Here's an example of the addition operator in the WHERE clause using the data in the following employee table:

employee_ID	**employee_name**	**salary**	**allowance**
1	Alex	24000	1000
2	John	55000	1000
3	James	52000	1000
4	Sam	24000	1000

Let's say you want to retrieve the salaries of employees whose total salary is 25000. This is how you can use the addition operator in SQL:

```
SELECT *

FROM employee

WHERE salary + allowance = 25000;
```

The SQL statement filters the records of all employees whose total salary (salary plus allowance) is 25000.

The output displays the records of two employees with ID 1 and 4 as in the table below.

employee_ID	**employee_name**	**salary**	**allowance**
1	Alex	24000	1000
4	Sam	24000	1000

Using the subtraction operator

The SQL subtraction operator performs mathematical subtraction on numerical data within columns in a database table. If you want to subtract the values of one numerical column from the values of another numerical column, you must specify both columns as the first and second operands along with the subtraction operator. The syntax is as follows:

```
SELECT column_name1 - column_name2 FROM table_name;
```

Let's examine an example. Here's the employee table once again, but this time with a "Tax" column and several instances of new data.

employee_ID	**employee_name**	**salary**	**allowance**	**tax**
1	Alex	24000	1000	1000

employee_ID	employee_name	salary	allowance	tax
2	John	55000	1000	2000
3	James	52000	1000	2000
4	Sam	24000	1000	1000

Let’s say you want to retrieve the salaries of employees after deducting tax. This is the SQL syntax that you can use with the subtraction operator to get these results.

```
SELECT salary - tax FROM employee;
```

Here, the salary and tax columns are the operands, and the subtraction operator is applied to them. The values in the tax column are deducted from the values in the salary column. The output is as follows:

salary – tax
23000
53000
50000
23000
23000

Here’s an example of using the subtraction operator in the WHERE clause. Consider the same employee table and data. If you want to find out who earns a salary of 50000 after the tax deduction, this is the SQL query you can write:

```
SELECT *

FROM employee

WHERE salary - tax = 50000;
```

Here, you are filtering out employees who receive a salary of 50000 after tax in the WHERE clause. This is the result that you would get.

employee_ID	employee_name	salary	allowance	tax
3	James	52000	1000	2000

Using the multiplication operator

The SQL multiplication operator performs the mathematical multiplication operation on the numerical data typed columns in a database table. If you want to multiply the values of two numerical columns, you must specify both columns as the first and second operand with the multiplication operator between them.

Let’s say in the employee table, you want to generate the tax amounts for each employee if these amounts are doubled.

You would write a SQL SELECT statement like this.

SELECT tax * 2 FROM employee;

Here, you are doubling the tax for all employees by multiplying the tax column value by 2.

The result would be as follows:

tax * 2
2000
4000
4000
2000

Now let's review an example of how to use the multiplication operator in the WHERE clause.Let's say you want to know who must pay an amount of tax equal to 4000, after doubling the current tax value.
The SELECT query gives the desired result, using the multiplication operator in the WHERE clause.

```
SELECT *

FROM employee

WHERE tax * 2 = 4000;
```

Here, the WHERE clause filters out the employees' records. It shows who'll be paying tax amounting to 4000 after the current tax amount is doubled. The result is as follows:

employee_ID	**employee_name**	**salary**	**allowance**	**tax**
2	John	55000	1000	2000
3	James	52000	1000	2000

Using the division operator
The division operator divides the numerical values of one column by the numerical values of another column. The syntax of using the division operator is as follows:
SELECT column_name1 Division_Operator column_name2 FROM table_name;

The data in the employee table is as follows:

employee_id	**employee_name**	**salary**	**allowance**	**tax**
1	alex	24000	1000	1000
2	John	55000	3000	2000
3	James	52000	3000	2000
4	Sam	24000	1000	1000

In this next example, let's say that you want to find out the allowance percentage each employee receives, by using the salary and the allowance amount.
You can write a SQL SELECT statement with the division operator as follows:

```
SELECT allowance / salary * 100 FROM employee;
```

Here, both the division and multiplication operators are used together to divide the allowance by salary and multiply the answer by 100 to find out the allowance percentage.
The result is as follows:

allowance / salary * 100
4.1667
5.4545
5.7692
4.1667

Like the other arithmetic operators, this too can be used in the WHERE clause of a SELECT statement.Let's say you want to find out which of the employees get an allowance of at least 5%.
Here's how the division operation is used in the WHERE clause to achieve this:

```
SELECT *

FROM employee

WHERE allowance / salary * 100 >= 5;
```

Both the division and the multiplication operators are used in the WHERE clause to filter out the employees who receive an allowance of at least 5% of their salary. The output is as follows:

employee_ID	employee_name	salary	allowance	tax
2	John	55000	3000	2000
3	James	52000	3000	2000

Using the modulus operator
The modulus operator (%) behaves as it's expected in SQL by giving the remainder when the numerical values of one column is divided by the numerical values of another column. The syntax is as follows:

```
SELECT column_name1 % column_name2 FROM table_name;
```

In this example, you're working with the following data in the employee table.

employee_ID	employee_name	salary	hours	allowance	tax
1	alex	24000	10	1000	1000
2	John	55000	11	3000	2000
3	James	52000	7	3000	2000
4	Sam	24000	11	1000	1000

You want to find out if the number of hours worked by each employee is an even number. You can find this out by issuing the following SQL statement:

```
SELECT hours % 2 FROM employee;
```

Here, the modulus operator is applied to the hours column to see if there’s a remainder when it’s divided by 2. If the remainder is zero, that means the number of hours worked by that employee is an even number. If the remainder is greater than 0, that means the number of hours worked by that employee is an odd number.
In this case, the output is as follows:

salary % hours
0
1
1
1

As the output shows, only employee 1 has worked for an even number of hours. The rest of the employees have worked for an odd number of hours. If you want to filter out only the employees who worked for an even number of hours, use the following SELECT statement that uses the modulus operator in the WHERE clause:

```sql
SELECT *

FROM employee

WHERE hours % 2 = 0;
```

The result of this SQL statement shows the employee record with id 1 as shown in the following table:

employee_ID	employee_name	salary	hours	allowance	tax
1	alex	24000	10	1000	1000

In this reading, you learned more about the use of arithmetic operators in SQL including addition, subtraction, multiplication, division, and modulus. The examples given should help you to understand how these operators can be used in the SELECT and the WHERE clauses.