OPERATORS IN SQL

SQL Series Part 5

-Mayuri Dandekar

WHAT IS OPERATORS?

The SQL **reserved words and characters** are called operators, which are **used with a WHERE** clause in a SQL query.

Most used operators:

- **Arithmetic operators** : arithmetic operations on numeric values

  Example: Addition (+), Subtraction (-), Multiplication (*), Division (/), Modulus (%)

- **Comparison operators**: compare two different data of SQL table

  Example: Equal (=), Not Equal (!=), Greater Than (>), Greater Than Equals to (>=)

- **Logical operators**: perform the Boolean operations

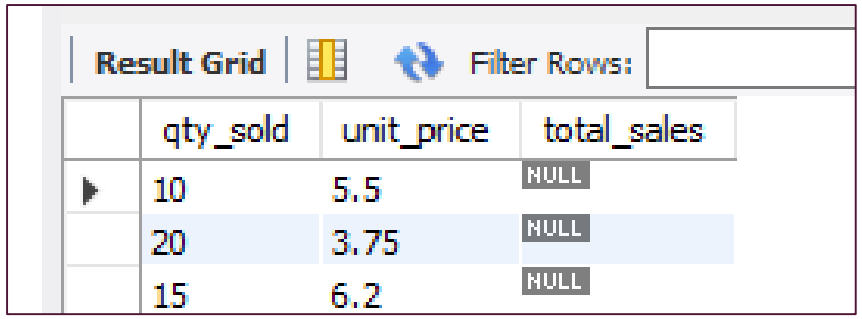
  Example: ALL, IN, BETWEEN, LIKE, AND, OR, NOT, ANY

- **Bitwise operators**: perform the bit operations on the Integer values

  Example: Bitwise AND (&), Bitwise OR(|)

ARITHMETIC OPERATOR -- ADDITION

Adds two numeric values.



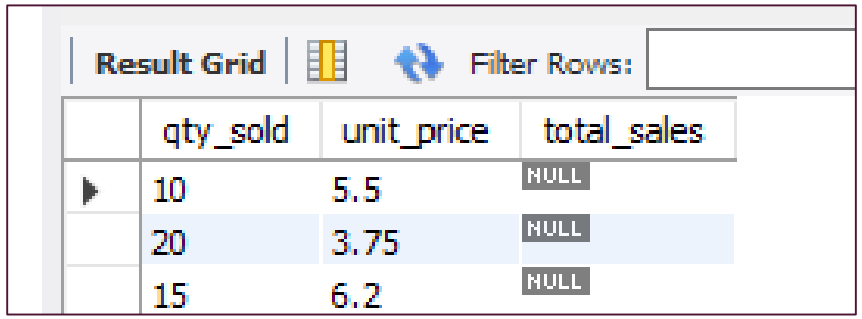

```
2 •   SELECT unit_price + 10 AS addition
3         FROM sales_data;
4
```

Result Grid | Filter Rows: | Export: | Wrap

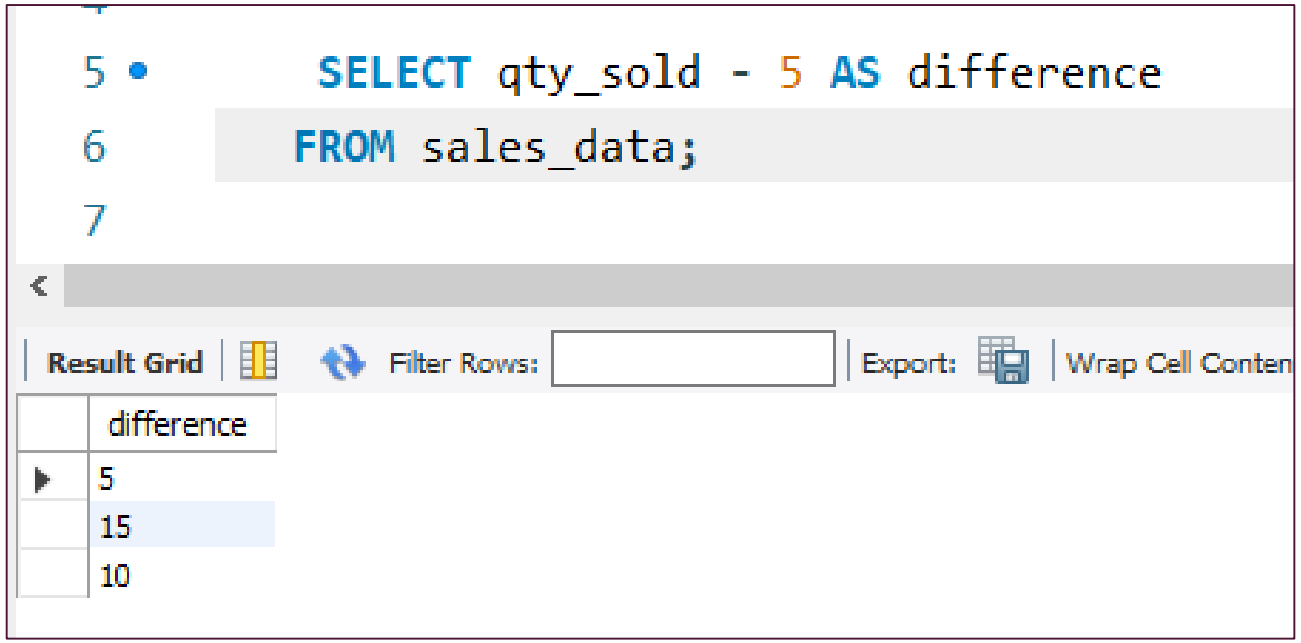
addition
15.5
13.75
16.199999809265137

ARITHMETIC OPERATOR -- SUBTRACTION

Subtracts one numeric value from another.

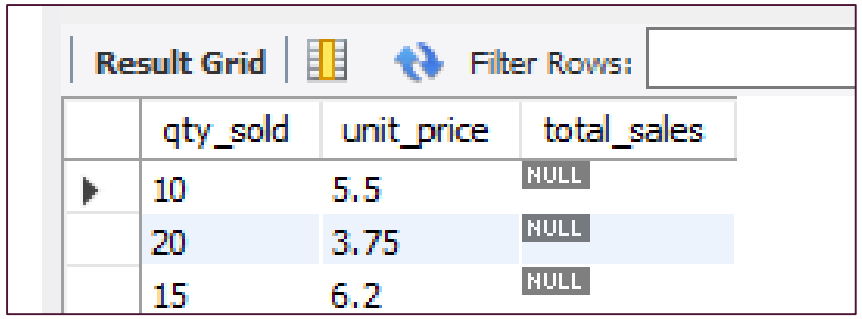



```
5 •   SELECT qty_sold - 5 AS difference
6     FROM sales_data;
7
```



ARITHMETIC OPERATOR -- MULTIPLICATION

Multiplies two numeric values.



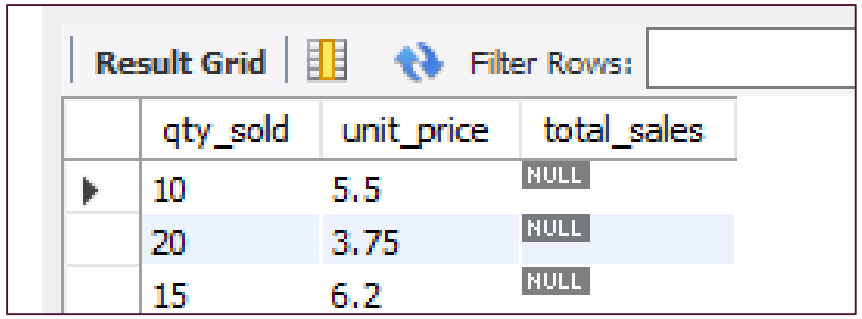

```
8 •  set sql_safe_updates=0;
9 •  UPDATE sales_data
10      SET total_sales = qty_sold * unit_price;
11 •  select * from sales_data;
12
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

qty_sold	unit_price	total_sales
10	5.5	55
20	3.75	75
15	6.2	93

ARITHMETIC OPERATOR -- DIVISION

Divides one numeric value by another.




```
13 •   UPDATE sales_data
14         SET total_sales = total_sales / 2;
15
```

Result Grid | Filter Rows: | Export: | Wrap Cel

qty_sold	unit_price	total_sales
10	5.5	27.5
20	3.75	37.5
15	6.2	46.5

ARITHMETIC OPERATOR -- MODULUS

Returns the remainder of a division operation.

Result Grid | Filter Rows:

	qty_sold	unit_price	total_sales
▶	10	5.5	NULL
	20	3.75	NULL
	15	6.2	NULL

```
16 •    UPDATE sales_data
17      SET total_sales = total_sales % 20;
18
```

Result Grid | Filter Rows: | Export: | Wrap Cell Cor

	qty_sold	unit_price	total_sales
▶	10	5.5	7.5
	20	3.75	17.5
	15	6.2	6.5

COMPARISON OPERATOR – EQUAL TO(=)

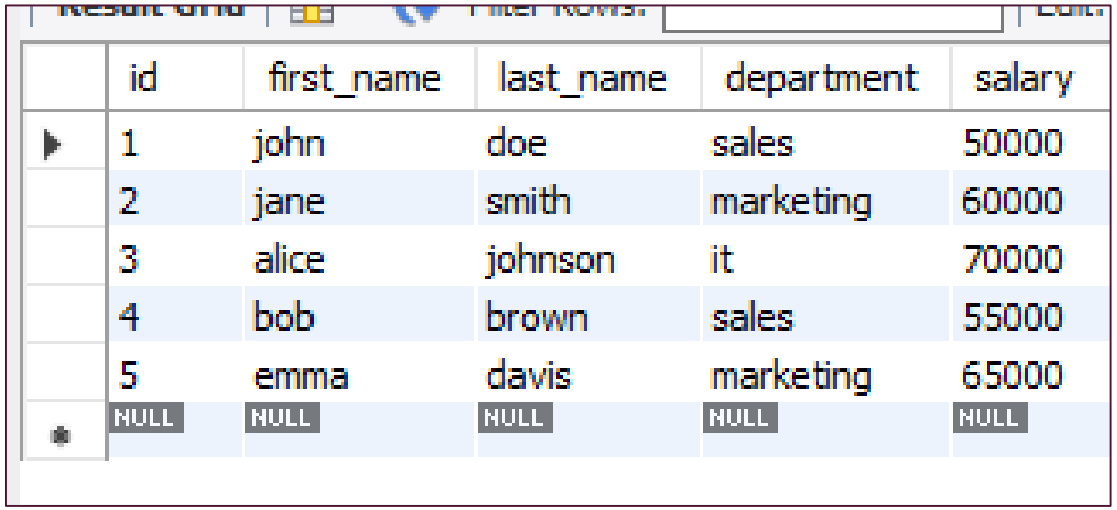
id	first_name	last_name	department	salary
1	john	doe	sales	50000
2	jane	smith	marketing	60000
3	alice	johnson	it	70000
4	bob	brown	sales	55000
5	emma	davis	marketing	65000
NULL	NULL	NULL	NULL	NULL

```
40 •   SELECT *
41         FROM employees
42         WHERE department = 'Sales';
```

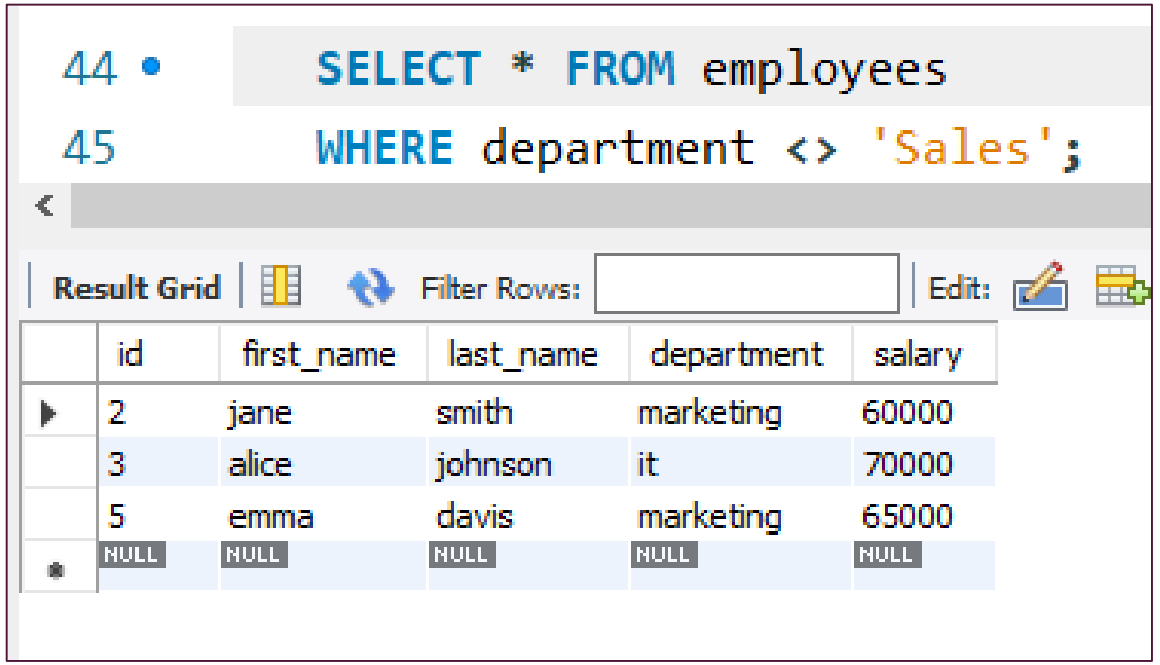
Result Grid | Filter Rows: | Edit:

id	first_name	last_name	department	salary
1	john	doe	sales	50000
4	bob	brown	sales	55000
NULL	NULL	NULL	NULL	NULL

COMPARISON OPERATOR – NOT EQUAL TO(<> OR !=)



id	first_name	last_name	department	salary
1	john	doe	sales	50000
2	jane	smith	marketing	60000
3	alice	johnson	it	70000
4	bob	brown	sales	55000
5	emma	davis	marketing	65000
NULL	NULL	NULL	NULL	NULL



```
44 •    SELECT * FROM employees
45      WHERE department <> 'Sales';
```

Result Grid | Filter Rows: | Edit:

id	first_name	last_name	department	salary
2	jane	smith	marketing	60000
3	alice	johnson	it	70000
5	emma	davis	marketing	65000
NULL	NULL	NULL	NULL	NULL

COMPARISON OPERATOR – GREATER THAN(>)

id	first_name	last_name	department	salary
1	john	doe	sales	50000
2	jane	smith	marketing	60000
3	alice	johnson	it	70000
4	bob	brown	sales	55000
5	emma	davis	marketing	65000
NULL	NULL	NULL	NULL	NULL

```
47 •    SELECT * FROM employees
48      WHERE salary > 60000;
49
```

Result Grid | Filter Rows: | Edit:

id	first_name	last_name	department	salary
3	alice	johnson	it	70000
5	emma	davis	marketing	65000
NULL	NULL	NULL	NULL	NULL

COMPARISON OPERATOR – LESS THAN(<)

id	first_name	last_name	department	salary
1	john	doe	sales	50000
2	jane	smith	marketing	60000
3	alice	johnson	it	70000
4	bob	brown	sales	55000
5	emma	davis	marketing	65000
NULL	NULL	NULL	NULL	NULL

```
50 •    SELECT * FROM employees
51      WHERE salary < 60000;
```

Result Grid | Filter Rows: | Edit:

id	first_name	last_name	department	salary
1	john	doe	sales	50000
4	bob	brown	sales	55000
NULL	NULL	NULL	NULL	NULL

COMPARISON OPERATOR – GREATER THAN OR EQUAL TO (>=)

id	first_name	last_name	department	salary
1	john	doe	sales	50000
2	jane	smith	marketing	60000
3	alice	johnson	it	70000
4	bob	brown	sales	55000
5	emma	davis	marketing	65000
NULL	NULL	NULL	NULL	NULL

```
53 •    SELECT * FROM employees
54      WHERE salary >= 60000;
55
```

Result Grid | Filter Rows: | Edit:

id	first_name	last_name	department	salary
2	jane	smith	marketing	60000
3	alice	johnson	it	70000
5	emma	davis	marketing	65000
NULL	NULL	NULL	NULL	NULL

COMPARISON OPERATOR – LESS THAN OR EQUAL TO (<=)

id	first_name	last_name	department	salary
1	john	doe	sales	50000
2	jane	smith	marketing	60000
3	alice	johnson	it	70000
4	bob	brown	sales	55000
5	emma	davis	marketing	65000
NULL	NULL	NULL	NULL	NULL

```
56 •    SELECT * FROM employees
57      WHERE salary <= 60000;
58
```

Result Grid | Filter Rows: | Edit:

id	first_name	last_name	department	salary
1	john	doe	sales	50000
2	jane	smith	marketing	60000
4	bob	brown	sales	55000
NULL	NULL	NULL	NULL	NULL

LOGICAL OPERATOR – AND

id	first_name	last_name	department	salary
1	john	doe	sales	50000
2	jane	smith	marketing	60000
3	alice	johnson	it	70000
4	bob	brown	sales	55000
5	emma	davis	marketing	65000
NULL	NULL	NULL	NULL	NULL

```
61 •    SELECT * FROM employees
62      WHERE department = 'Sales' AND salary > 50000;
```

Result Grid | Filter Rows: | Edit: | Export/Import:

id	first_name	last_name	department	salary
4	bob	brown	sales	55000
NULL	NULL	NULL	NULL	NULL

LOGICAL OPERATOR – OR

id	first_name	last_name	department	salary
1	john	doe	sales	50000
2	jane	smith	marketing	60000
3	alice	johnson	it	70000
4	bob	brown	sales	55000
5	emma	davis	marketing	65000
NULL	NULL	NULL	NULL	NULL

```
64 •    SELECT * FROM employees
65      WHERE department = 'Sales' OR
66      department = 'Marketing';
67
```

Result Grid | Filter Rows: | Edit:

id	first_name	last_name	department	salary
1	john	doe	sales	50000
2	jane	smith	marketing	60000
4	bob	brown	sales	55000
5	emma	davis	marketing	65000
NULL	NULL	NULL	NULL	NULL

LOGICAL OPERATOR – NOT

id	first_name	last_name	department	salary
1	john	doe	sales	50000
2	jane	smith	marketing	60000
3	alice	johnson	it	70000
4	bob	brown	sales	55000
5	emma	davis	marketing	65000
NULL	NULL	NULL	NULL	NULL

```
68 •    SELECT * FROM employees
69      WHERE NOT department = 'Sales';
70
```

Result Grid | Filter Rows: | Edit:

id	first_name	last_name	department	salary
2	jane	smith	marketing	60000
3	alice	johnson	it	70000
5	emma	davis	marketing	65000
NULL	NULL	NULL	NULL	NULL

LOGICAL OPERATOR – ALL

id	first_name	last_name	department	salary
1	john	doe	sales	50000
2	jane	smith	marketing	60000
3	alice	johnson	it	70000
4	bob	brown	sales	55000
5	emma	davis	marketing	65000
NULL	NULL	NULL	NULL	NULL

```
71    -- salary greater than all salaries in the Marketing dept--
72 •  SELECT * FROM employees
73    WHERE salary > ALL
74    (SELECT salary FROM employees WHERE department = 'Marketing');
75
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

id	first_name	last_name	department	salary
3	alice	johnson	it	70000
NULL	NULL	NULL	NULL	NULL

LOGICAL OPERATOR – IN

id	first_name	last_name	department	salary
1	john	doe	sales	50000
2	jane	smith	marketing	60000
3	alice	johnson	it	70000
4	bob	brown	sales	55000
5	emma	davis	marketing	65000
NULL	NULL	NULL	NULL	NULL

```
76 •  SELECT * FROM employees
77    WHERE department IN ('Sales', 'Marketing');
78
```

Result Grid | Filter Rows: | Edit: | Export/Impo

id	first_name	last_name	department	salary
1	john	doe	sales	50000
2	jane	smith	marketing	60000
4	bob	brown	sales	55000
5	emma	davis	marketing	65000
NULL	NULL	NULL	NULL	NULL

LOGICAL OPERATOR – BETWEEN

id	first_name	last_name	department	salary
1	john	doe	sales	50000
2	jane	smith	marketing	60000
3	alice	johnson	it	70000
4	bob	brown	sales	55000
5	emma	davis	marketing	65000
NULL	NULL	NULL	NULL	NULL

```
79 •  SELECT * FROM employees
80    WHERE salary BETWEEN 50000 AND 60000;
```

Result Grid | Filter Rows: | Edit:

id	first_name	last_name	department	salary
1	john	doe	sales	50000
2	jane	smith	marketing	60000
4	bob	brown	sales	55000
NULL	NULL	NULL	NULL	NULL

LOGICAL OPERATOR – LIKE

id	first_name	last_name	department	salary
1	john	doe	sales	50000
2	jane	smith	marketing	60000
3	alice	johnson	it	70000
4	bob	brown	sales	55000
5	emma	davis	marketing	65000
NULL	NULL	NULL	NULL	NULL

```
82 •   SELECT * FROM employees
83     WHERE first_name LIKE 'J%';
```

Result Grid | Filter Rows: | Edit:

id	first_name	last_name	department	salary
1	john	doe	sales	50000
2	jane	smith	marketing	60000
NULL	NULL	NULL	NULL	NULL

LOGICAL OPERATOR – ANY

id	first_name	last_name	department	salary
1	john	doe	sales	50000
2	jane	smith	marketing	60000
3	alice	johnson	it	70000
4	bob	brown	sales	55000
5	emma	davis	marketing	65000
NULL	NULL	NULL	NULL	NULL

```
85    -- salary greater than any salary in the Marketing dept --
86 •  SELECT * FROM employees
87    WHERE salary > ANY
88    (SELECT salary FROM employees WHERE department = 'Marketing');
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

id	first_name	last_name	department	salary
3	alice	johnson	it	70000
5	emma	davis	marketing	65000
NULL	NULL	NULL	NULL	NULL

BITWISE OPERATOR – AND(&)

Result Grid | Filter Rows:

	qty_sold	unit_price	total_sales
▶	10	5.5	NULL
	20	3.75	NULL
	15	6.2	NULL

```
91 •  SELECT qty_sold & unit_price AS result
92    from sales_data;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Cor

	result
▶	2
	4
	6

BITWISE OPERATOR – OR(|)

Result Grid | Filter Rows:

	qty_sold	unit_price	total_sales
▶	10	5.5	NULL
	20	3.75	NULL
	15	6.2	NULL

```
94 •  SELECT qty_sold | unit_price AS result
95    from sales_data;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Cor

	result
▶	14
	20
	15

THANK YOU!!!

SQL Series Part 5

-Mayuri Dandekar