Arithmetic Operator

Arithmetic operators are:

+	[Addition]
-	[Subtraction]
/	[Division]
*	[Multiplication]
%	[Modulus]

Scenario 1: write a query to add two numbers using addition Operator

```
SELECT 1100 + 1900;
```

Output:

1100 + 1900
3000

Scenario 2: write a query to subtract two numbers using subtraction Operator

```
SELECT 1933 - 1288;
```

Output:

1933 - 1288
645

Scenario 3: write a query to multiply two numbers using multiplication Operator

```
SELECT 1200 * 1200;
```

Output:

1200 * 1200
1440000

Scenario 4: write a query to divide two numbers using Division Operator

```
SELECT 1200/500;
```

Output:

1200 / 500
2.4000

Scenario 5: write a query to get reminder of two numbers using modulus Operator

```
SELECT 11%2;
```

Output:

11%2

1

To change the column name you can make use of aliasing as shown below

```
SELECT 11%2 as result;
```

Output:

result
1

Scenario 6: write a query for the expression **100/10*2+5-2%3**

```
SELECT 100/10*2+5-2%3 as result;
```

Here whenever you are working on the expression it will look for precedent and precedence looks as shown below

Precedence
%
*, /
+, -

Output:

result
23.0000

Now let us consider the department and employee table to understand the arithmetic operator

Department table

dept_id	**dept_name**	**mrg_id**
20	sales	8
30	executive	4
40	shipping	6
50	marketing	3
70	accounting	2
80	IT	1

Employee table:

emp_id	**first_name**	**last_name**	**email**	**hire_date**	**salary**	**dept_id**
1	kelly	davis	davis@gmail.com	2021-01-22	78000	80
2	tom	taylor	tom@gmail.com	2020-09-22	84200	70
3	mike	whalen	mike@gmail.com	2021-06-30	98200	50
4	andy	lumb	andy@gmail.com	2021-02-27	42200	80
5	anjel	nair	anj@gmail.com	2019-09-26	42200	40
6	ram	kumar	ram@gmail.com	2018-12-26	64200	40

7	rohan	sharma	ro@gmail.com	2018-11-24	84200	20
8	john	king	j0@gmail.com	2021-02-09	124200	20

Scenario 1: Display first name, last name and salary of all the employees by adding 1000 to the salary of all the employees

```sql
SELECT
first_name as fn, last_name as ln, salary+1000 as salary
FROM
employee;
```

Output:

fn	ln	salary
kelly	davis	79000
tom	taylor	85200
mike	whalen	99200
andy	lumb	43200
anjel	nair	43200
ram	kumar	65200
rohan	sharma	85200
john	king	125200

Scenario 2: Display first name, last name and salary of all the employees by adding dept_id to the salary of all the employees

```
SELECT
first_name as fn, last_name as ln, salary+dept_id as salary
FROM
employee;
```

Output:

fn	ln	salary
kelly	davis	78080
tom	taylor	84270
mike	whalen	98250
andy	lumb	42280
anjel	nair	42240

ram	kumar	64240
rohan	sharma	84220
john	king	124220

Scenario 3: Display first name, last name and salary of all the employees by subtracting 1000 to the salary of all the employees

```
SELECT
first_name as fn, last_name as ln, salary-1000 as salary
FROM
employee;
```

Output:

fn	ln	salary
kelly	davis	77000
tom	taylor	83200
mike	whalen	97200
andy	lumb	41200
anjel	nair	41200
ram	kumar	63200
rohan	sharma	83200
john	king	123200

Scenario 4: Display first name, last name and annual salary of all the employees.

annual salary = salary * 12

```
SELECT
first_name as fn, last_name as ln, salary*12 as annula_salary
FROM
employee;
```

Output:

fn	ln	salary
kelly	davis	936000
tom	taylor	1010400
mike	whalen	1178400
andy	lumb	506400
anjel	nair	506400
ram	kumar	770400
rohan	sharma	1010400
john	king	1490400

Scenario 5: Display first name, last name and annual salary of all the employees with dept_id as 20

```
SELECT
first_name as fn, last_name as ln, salary*12 as annual_salary
FROM
employee
WHERE
dept_id =20;
```

Output:

fn	**ln**	**annual_salary**
rohan	sharma	1010400
john	king	1490400

Scenario 6: Display first name, last name and half year salary of all the employees

```
SELECT
first_name as fn, last_name as ln, salary*12/2 as annual_salary
FROM
employee;
```

Output:

fn	**ln**	**salary**
kelly	davis	468000.0000
tom	taylor	505200.0000
mike	whalen	589200.0000
andy	lumb	253200.0000
anjel	nair	253200.0000
ram	kumar	385200.0000
rohan	sharma	505200.0000
john	king	745200.0000

Scenario 7: Display first name, last name and annual, half year, quarterly salary of all the employees

```
SELECT
first_name as fn, last_name as ln, (salary * 12) as
annual_salary, (salary*12/2) as half_yearly_salary,
(salary*12/4) as quarterly_salary
FROM
employee;
```

Output:

first_name	**last_name**	**annual_salary**	**half_yearly_salary**	**quarterly_salary**
kelly	davis	936000	468000.0000	234000.0000
tom	taylor	1010400	505200.0000	252600.0000
mike	whalen	1178400	589200.0000	294600.0000
andy	lumb	506400	253200.0000	126600.0000
anjel	nair	506400	253200.0000	126600.0000
ram	kumar	770400	385200.0000	192600.0000
rohan	sharma	1010400	505200.0000	252600.0000
john	king	1490400	745200.0000	372600.0000

Scenario 8: Display first name, last name and salary of all the employees where salary is not divisible(reminder is 0) by 3

```
SELECT
first_name as fn, last_name as ln, salary
FROM
employee
WHERE
salary%3 != 0;
```

Output:

fn	ln	salary
tom	taylor	84200
mike	whalen	98200
andy	lumb	42200
anjel	nair	42200
rohan	sharma	84200

Scenario 9: Display first name, last name and salary of all the employees with salary increment by 10%

```
SELECT first_name as fn, last_name as ln,
salary+(salary*10/100) FROM  sql_notes.employee;
```

Output:

fn	ln	salary
kelly	davis	85800.0000
tom	taylor	92620.0000

mike	whalen	108020.0000
andy	lumb	46420.0000
anjel	nair	46420.0000
ram	kumar	70620.0000
rohan	sharma	92620.0000
john	king	136620.0000