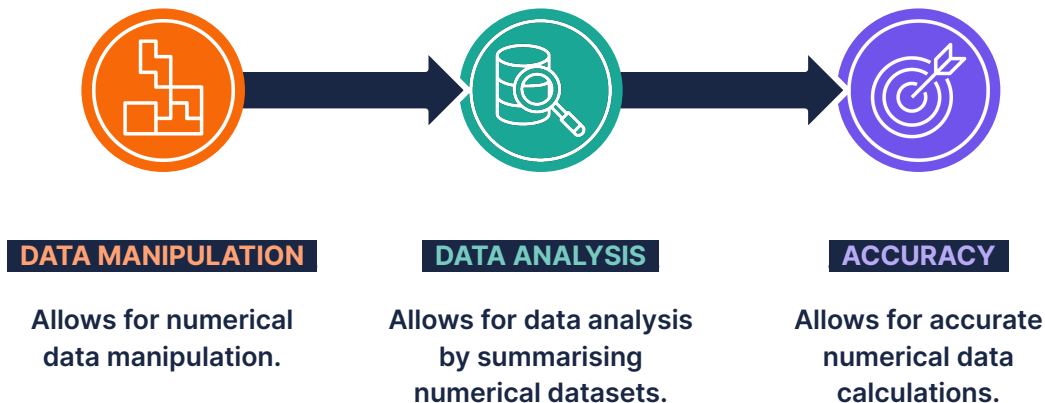


SQL numeric functions

Numeric functions in SQL

The importance of numeric functions in SQL

Numeric functions are **built-in functions that operate on numeric data types** (such as integers, decimals, and floating-points) and perform various **mathematical and statistical operations** on them.



Introduction to numeric functions in SQL

Aggregate numeric functions

- MIN()
- MAX()
- AVG()
- SUM()
- COUNT()

Scalar numeric functions

- ROUND()
- SQRT()
- LOG()



Scalar numeric functions are not limited to the above. These will be discussed in this section.

(General Household Survey (GHS) dataset

Province	Y_2018	Y_2019	Y_2020	Y_2021	Y_2022
Eastern_cape	23612.117	24218.898	25785.763	27074.594	28049.845
Free_state	25974.478	27897.917	29459.302	30932.342	48532.912
Gauteng	31748.937	33112.344	34966.168	36715.740	38378.338
Kwazulu_natal	26188.484	27496.383	29141.264	30598.355	31914.132
Limpopo	30140.911	31726.672	33502.167	35178.147	36820.465
Mpumalanga	27321.994	28283.472	29867.734	31360.122	32963.496
North_west	25522.502	27621.283	29168.121	30626.714	32500.753
Northern_cape	27833.420	29288.380	30929.224	32475.897	33222.342
Western_cape	30164.275	31863.417	33647.873	35329.837	36919.034

This dataset shows the **average monthly salaries of public servants** obtained from 2018 to 2022 by province.

We will use it to illustrate **different numeric functions**.

Name: Salaries.ghs_db

The MIN() function

The MIN() function returns the **smallest or lowest value** of the selected column. Suppose we want to calculate the **lowest average monthly salary** made in the year 2020.

Query

```
SELECT  
    MIN(Y_2020) AS Lowest_salary_2020  
FROM  
    Salaries.ghs_db;
```

Output

Lowest_salary_2020
25789.763

Y_2020
25785.763
29459.302
34966.168
29141.264
33502.167
29867.734
29168.121
30929.224
33647.873

The MAX() function

The MAX() function returns the **largest or highest value** of the selected column. Suppose we want to calculate the **highest average monthly salary** made in the same year, 2020.

Query

```
SELECT  
    MAX(Y_2020) AS Highest_salary_2020  
FROM  
    Salaries.ghs_db;
```

Output

Highest_salary_2020

34966.168

Y_2020
25785.763
29459.302
34966.168
29141.264
33502.167
29867.734
29168.121
30929.224
33647.873

The AVG() function

The `AVG()` function returns the **average value** of the selected numeric column. Suppose we want to calculate the **average monthly salary** made in the year 2022.

Query

```
SELECT
    AVG(Y_2022) AS Average_salary_2022
FROM
    Salaries.ghs_db;
```

Output

Average_salary_2022

35477.92419

Y_2022

28049.845

48532.912

38378.338

31914.132

36820.465

32963.496

32500.753

33222.342

36919.034

The SUM() function

The SUM() function returns the **total sum** of a specified numeric column. Suppose we want to find the sum of monthly salaries made in the year 2019.

Query

```
SELECT
    SUM(Y_2019) AS Total_salaries_2022
FROM
    Salaries.ghs_db;
```

Output

Total_salaries_2019

319301.3177

Y_2019

24218.898

27897.917

33112.344

27496.383

31726.672

28283.472

27621.283

29288.380

31863.417

The COUNT() function

The COUNT() function returns the **number of rows** of a specified column. Suppose we want to know how many provinces were used in this dataset.

Query

```
SELECT
    COUNT(Province) AS Number_of_provinces
FROM
    Salaries.ghs_db;
```

Output

Number_of_provinces
9

Province
eastern_cape
free_state
gauteng
kwazulu_natal
limpopo
mpumalanga
north_west
northern_cape
western_cape

The COUNT(DISTINCT column) function

The COUNT(DISTINCT column) function returns the distinct or unique **number of rows** of a specified column. Suppose we want to know how many unique provinces were used in this dataset.

Query

```
SELECT  
    COUNT(DISTINCT Province) AS Number_of_provinces  
FROM  
    Salaries.ghs_db;
```

Output

Number_of_provinces
9

The unique number of provinces is equal to the number we got from the previous example. This is because there are no duplicate provinces in the Provinces column.

The ROUND() function

The ROUND() function **rounds a numerical value to a specified number of decimal places**. The syntax is as follows:

```
SELECT ROUND( numerical value , decimal places )
```

This clause specifies the numerical value to be rounded.

This clause refers to the number of decimal places to round the numerical value to.



The ROUND() function

Suppose we want to round the **average monthly salary** made in the year 2022 we calculated before to the nearest cent.

Query

```
SELECT
    ROUND(Y_2022,2) AS Rounded_salaries
FROM
    Salaries.ghs_db;
LIMIT 3
```

Output

Rounded_salaries
23612.117
25974.478
31748.937

Y_2022	Y_2022
23612.12	23612.117
25974.48	25974.478
31748.94	31748.937
26188.48	26188.484
30140.91	30140.911
27321.99	27321.994
25522.50	25522.502
27833.42	27833.420
30164.28	30164.275

The SQRT() function

The SQRT() function returns **the square root of a numerical value**. The syntax is as follows:

```
SELECT SQRT( numerical value )
```

This clause specifies the numerical value to be square rooted.

The SQRT() function

The SQRT() function returns **the square root of a numerical value**. The syntax is as follows:

Query

```
SELECT
    SQRT(Y_2021) AS Square_root_y_2021
FROM
    Salaries.ghs_db;
LIMIT 3;
```

Output

Square_root_y_2021
160.5794607...
171.6371221...
186.9924267...

Y_2021	Y_2021
160.58...	27074.594
171.64...	30932.342
186.99...	36715.740
170.71...	30598.355
183.04...	35178.147
172.82...	31360.122
170.79...	30626.714
175.87...	32475.897
183.43...	35329.837

The LOG() function

The LOG(base, numerical) function returns the **logarithm of a numeric value with a specified base**. The syntax is as follows:

```
SELECT LOG( numerical_value, base );
```

This clause specifies the numerical value to be logged.

This clause specifies the base of the logarithm of the numerical value.

The LOG() function

Suppose we want to find the log of the average salaries and then round it to 2 decimal places.

Query

```
SELECT
    LOG(Y_2021,10) AS Log_y_2021
FROM
    salaries.ghs_db;
LIMIT 3;
```

Output

Log_y_2021
4.564852289...
4.485698078...
4.546272965...

Y_2021	Y_2021`
4.564852289	27074.594
4.485698078	30932.342
4.546272965	36715.740
4.496377744	30598.355
4.486100396	35178.147
4.511561155	31360.122
4.548141639	30626.714
4.564852289	32475.897
4.485698078	35329.837