

The **NTILE()** function is a powerful window function in SQL that allows you to divide an ordered set of rows into a specified number of "tiles" or groups. In your selected query:

- **NTILE(4)**: This specifies that the cities are to be divided into 4 groups, meaning you will get values 1, 2, 3, or 4 for Population Quartile.
- **PARTITION BY Country Code**: This is crucial because it tells the function to perform this division independently for each Country Code. So, instead of dividing *all* cities in the database into 4 quartiles, it divides cities *within each country* into 4 quartiles.
- **ORDER BY Population ASC**: Within each country, the cities are ordered by their Population in ascending order (from least to most populated). The NTILE() function then assigns the lowest population cities to quartile 1, the next set to quartile 2, and so on.

How to Interpret the Resulting PopulationQuartile:

- If a city has PopulationQuartile = 1, it means it falls into the **lowest 25%** of cities by population within its specific country.
- If PopulationQuartile = 2, it's in the **second 25%** (26-50%) of cities by population within that country.
- If PopulationQuartile = 3, it's in the **third 25%** (51-75%) of cities by population within that country.
- If PopulationQuartile = 4, it's in the **highest 25%** (76-100%) of cities by population within that country.

This function is very useful for segmenting data into equal-sized buckets based on a specific ordering, helping you analyse relative performance or distribution within defined groups.

The screenshot shows a SQL query being run in a database environment. The query uses the NTILE(4) window function to divide cities within each country into four population quartiles. The results are displayed in a grid format below the query text.

```
1 /* This query uses the NTILE() window function to divide cities within each country into 4 population quartiles (tiles).
2 The cities are first partitioned by their CountryCode, and then ordered by their Population in ascending order within each country.*/
3 • USE World;
4 • SELECT
5     ID,
6     Name AS CityName,
7     CountryCode,
8     Population,
9     NTILE(4) OVER (
10         PARTITION BY CountryCode
11         ORDER BY Population ASC
12     ) AS PopulationQuartile
13 FROM
14     City
15 ORDER BY
16     CountryCode, PopulationQuartile, Population;
```

ID	CityName	CountryCode	Population	PopulationQuartile
129	Oranjestad	ABW	29034	1
4	Mazar-e-Sharif	AFG	127800	1
3	Herat	AFG	186800	2
2	Qandahar	AFG	237500	3
1	Kabul	AFG	1780000	4
60	Namibe	AGO	118200	1
59	Benguela	AGO	128300	1
58	Lobito	AGO	130000	2