Assignment Title:

Aggregation in SQL

Solution explanation

Submitted by:

Name: Sachin Brijesh Yadav

Course: Data Analytics With AI

Batch: September 2025 Live (Pro Batch)

Registered Email ID: sachinyadav9496@gmail.com

1. Count how many cities are there in each country

Query:

```sql
SELECT CountryCode, COUNT(*) AS city_count
FROM city
GROUP BY CountryCode;
```

Explanation:

This query groups all cities by their *CountryCode*.
Since each row in the `city` table represents one city, applying `COUNT(*)` gives the total number of cities in each country.
Using `GROUP BY` ensures that the counting happens separately for each country.

2. Display all continents having more than 30 countries

Query:

```sql
SELECT Continent, COUNT(*) AS country_count
FROM country
GROUP BY Continent
HAVING COUNT(*) > 30;
```

Explanation:

The `country` table contains one row per country.
Grouping by **Continent** counts how many countries belong to each continent.
The `HAVING` clause filters the grouped results to include only those continents where the number of countries is greater than 30.
This is a typical use of `HAVING` with aggregate functions.

3. List regions whose total population exceeds 200 million

Query:

```sql
SELECT Region, SUM(Population) AS total_population
FROM country
GROUP BY Region
HAVING SUM(Population) > 200000000;
```

Explanation:

The population of each country contributes to the total population of its region.
By grouping rows using `Region` and applying `SUM(Population)`, the query calculates the total population per region.
`HAVING` ensures only those regions are displayed where the population exceeds 200 million.

4. Find the top 5 continents by average GNP per country

Query:

```
SELECT Continent, AVG(GNP) AS avg_gnp
FROM country
GROUP BY Continent
ORDER BY avg_gnp DESC
LIMIT 5;
```

Explanation:

This query calculates the average Gross National Product (GNP) of countries within each continent using `AVG(GNP)`.
The results are sorted in descending order so that continents with the highest GNP come first.
`LIMIT 5` ensures that only the top five continents are displayed.

5. Total number of official languages spoken in each continent

Requires joining `country` and `countrylanguage`.

Query:

```
SELECT c.Continent, COUNT(cl.Language) AS official_languages
FROM country c
JOIN countrylanguage cl ON c.Code = cl.CountryCode
WHERE cl.IsOfficial = 'T'
GROUP BY c.Continent;
```

Explanation:

Each country can have multiple languages.
After joining `country` and `countrylanguage`, we filter only official languages using

`IsOfficial = 'T'`.
Then we group the data by continent to count how many official languages exist in each continent.
`COUNT(Language)` counts all official languages associated with countries in that continent.

6. Find maximum and minimum GNP for each continent

Query:

```
SELECT Continent,
       MAX(GNP) AS max_gnp,
       MIN(GNP) AS min_gnp
FROM country
GROUP BY Continent;
```

Explanation:

This query uses aggregate functions to determine the highest and lowest GNP values within each continent.
Grouping by continent ensures that `MAX` and `MIN` are calculated separately for every continent.

7. Find the country with the highest average city population

Requires joining `country` and `city`.

Query:

```
SELECT c.Name, AVG(ci.Population) AS avg_city_population
FROM country c
JOIN city ci ON c.Code = ci.CountryCode
GROUP BY c.Code
ORDER BY avg_city_population DESC
LIMIT 1;
```

Explanation:

Each country may contain multiple cities.
By joining the two tables and grouping by each country, the query calculates the average population of cities per country.
Sorting in descending order shows the country with the highest average city population.
`LIMIT 1` ensures only the top result is displayed.

8. List continents where average city population exceeds 200,000

Multi-level grouping: City → Country → Continent

Query:

```
SELECT c.Continent, AVG(ci.Population) AS avg_city_population
FROM country c
JOIN city ci ON c.Code = ci.CountryCode
GROUP BY c.Continent
HAVING AVG(ci.Population) > 200000;
```

Explanation:

City populations are linked to continents through the `country` table.
Using the join, we associate each city with its continent.
`AVG(ci.Population)` computes the average city population for each continent.
`HAVING` filters only those continents where the average exceeds 200,000.

9. Total population + average life expectancy per continent (sorted by life expectancy)

Query:

```
SELECT Continent,
       SUM(Population) AS total_population,
       AVG(LifeExpectancy) AS avg_life_expectancy
FROM country
GROUP BY Continent
ORDER BY avg_life_expectancy DESC;
```

Explanation:

Two aggregate functions operate here:

- `SUM(Population)` gives the total population of all countries in the continent.
- `AVG(LifeExpectancy)` gives the average life expectancy across those countries. Sorting descending places the continent with the highest life expectancy at the top.

10. Top 3 continents with highest average life expectancy, but only if total population exceeds 200 million

Combination of HAVING + ORDER + LIMIT

Query:

```
SELECT Continent,
       AVG(LifeExpectancy) AS avg_life_expectancy,
       SUM(Population) AS total_population
FROM country
GROUP BY Continent
HAVING SUM(Population) > 200000000
ORDER BY avg_life_expectancy DESC
LIMIT 3;
```

Explanation:

This query performs both **filtering** and **ranking**:

- `HAVING SUM(Population) > 200M` ensures only highly populated continents are considered.
- `AVG(LifeExpectancy)` is used to rank them by health outcomes.

`LIMIT 3` picks only the top three qualifying continents.