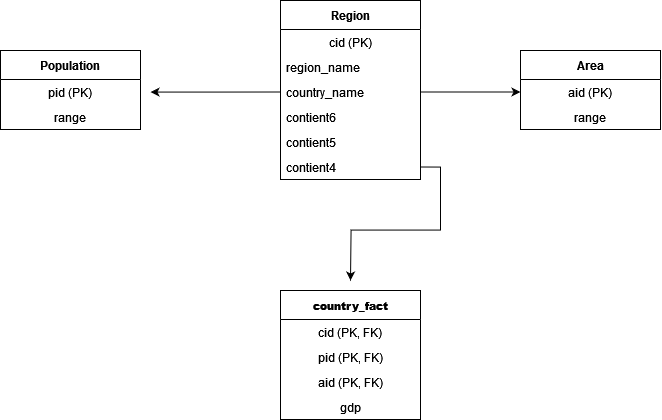
**1.** Create a STAR schema diagram of the country data warehouse. Indicate the primary key of each relation and if any, foreign keys

**Answer:**

**| **

**2.** Illustrate the hierarchy structure within the REGION dimension

**Answer:**

Africa (continent4)

├── Northern Africa (continent5)

│ ├── Maghreb (continent6)

│ │ ├── Algeria

│ │ ├── Morocco

│ │ ├── Tunisia

│ │ └── Libya

│ └── Nile Valley (continent6)

│ ├── Egypt

│ └── Sudan

├── Western Africa (continent5)

│ ├── Sahel (continent6)

│ │ ├── Mali

│ │ ├── Niger

│ │ ├── Chad

│ │ ├── Mauritania

│ │ └── Burkina Faso

│ └── Guinea Coast (continent6)

│ ├── Nigeria

│ ├── Ghana

│ ├── Ivory Coast

│ └── Senegal

└── Eastern Africa (continent5)

├── Horn of Africa (continent6)

│ ├── Ethiopia

│ ├── Somalia

│ ├── Eritrea

│ └── Djibouti

└── Great Lakes (continent6)

├── Kenya

├── Uganda

├── Tanzania

├── Rwanda

└── Burundi

**3.** Construct an SQL query to determine the average GDP per continent (labeled as continent5). Round the GPD value to no decimal places.

Query:

SELECT r.continent5,

ROUND(AVG(cf.gdp)) AS "Average GDP"

FROM country\_fact cf

JOIN region r ON cf.cid = r.cid

GROUP BY r.continent5;

Result:

A screenshot of a black and white screen

Description automatically generated

**4.** To explore the GDP within 'Euraisa', write an SQL query to **drill down** into the continent6 regions of ‘Euraisa’. Round the GPD value to no decimal places.

Query:

SELECT r.continent6 AS "Continent6 Name",

ROUND(AVG(cf.gdp)) AS "Average GDP"

FROM country\_fact cf

JOIN region r ON cf.cid = r.cid

WHERE r.continent5 = 'Eurasia'

GROUP BY r.continent6;

Result:

A screenshot of a computer

Description automatically generated

**5.** Develop an SQL query to assign a rank to each country in ‘America’ (where continent6 equals 'America') based on its GDP, in descending order.

Query:

SELECT

ROW\_NUMBER() OVER (PARTITION BY r.region\_name ORDER BY cf.gdp DESC) AS "RankNo",

r.region\_name AS "RegionName",

r.country\_name AS "CountryName",

cf.gdp AS "GDP"

FROM

country\_fact cf

JOIN

region r ON cf.cid = r.cid

WHERE

r.continent6 = 'America';

Result:

"RankNo" "RegionName" "CountryName" "GDP"

1 "Central America and the Caribbean" "Guatemala" 33000000000

2 "Central America and the Caribbean" "Puerto Rico" 26800000000

3 "Central America and the Caribbean" "Dominican Republic" 24000000000

4 "Central America and the Caribbean" "Costa Rica" 16900000000

5 "Central America and the Caribbean" "Trinidad and Tobago" 15000000000

6 "Central America and the Caribbean" "Cuba" 14000000000

7 "Central America and the Caribbean" "Panama" 12300000000

8 "Central America and the Caribbean" "El Salvador" 9800000000

9 "Central America and the Caribbean" "Honduras" 9700000000

10 "Central America and the Caribbean" "Jamaica" 7800000000

11 "Central America and the Caribbean" "Nicaragua" 6400000000

12 "Central America and the Caribbean" "Haiti" 5600000000

13 "Central America and the Caribbean" "The Bahamas" 4400000000

14 "Central America and the Caribbean" "Martinique" 3900000000

15 "Central America and the Caribbean" "Guadeloupe" 3800000000

16 "Central America and the Caribbean" "Barbados" 2400000000

17 "Central America and the Caribbean" "Netherlands Antilles" 1850000000

18 "Central America and the Caribbean" "Virgin Islands" 1200000000

19 "Central America and the Caribbean" "Aruba" 1100000000

20 "Central America and the Caribbean" "Cayman Islands" 700000000

21 "Central America and the Caribbean" "Saint Lucia" 610000000

22 "Central America and the Caribbean" "Belize" 575000000

23 "Central America and the Caribbean" "Antigua and Barbuda" 400000000

24 "Central America and the Caribbean" "Grenada" 258000000

25 "Central America and the Caribbean" "Saint Vincent and the Grenadines" 235000000

26 "Central America and the Caribbean" "Saint Kitts and Nevis" 210000000

27 "Central America and the Caribbean" "Dominica" 200000000

28 "Central America and the Caribbean" "British Virgin Islands" 133000000

29 "Central America and the Caribbean" "Turks and Caicos Islands" 80800000

30 "Central America and the Caribbean" "Montserrat" 55600000

31 "Central America and the Caribbean" "Anguilla" 49000000

32 "Central America and the Caribbean" "Navassa Island" 0

1 "North America" "United States" 6738400000000

2 "North America" "Mexico" 728700000000

3 "North America" "Canada" 639800000000

4 "North America" "Bermuda" 1700000000

5 "North America" "Saint Pierre and Miquelon" 66000000

1 "South America" "Brazil" 886300000000

2 "South America" "Argentina" 270800000000

3 "South America" "Venezuela" 178300000000

4 "South America" "Colombia" 172400000000

5 "South America" "Chile" 97700000000

6 "South America" "Peru" 73600000000

7 "South America" "Ecuador" 41100000000

8 "South America" "Uruguay" 23000000000

9 "South America" "Bolivia" 18300000000

10 "South America" "Paraguay" 15400000000

11 "South America" "Guyana" 1400000000

12 "South America" "Suriname" 1200000000

13 "South America" "French Guiana" 800000000

14 "South America" "Falkland Islands (Islas Malvinas)" 0

**6.** Develop an SQL query to first partition the countries within ‘America’ by region, and then, within each region, assign a rank to each country based on its GDP, in descending order.

Query:

SELECT

ROW\_NUMBER() OVER (PARTITION BY r.region\_name ORDER BY cf.gdp DESC) AS "RankNo",

r.region\_name AS "RegionName",

r.country\_name AS "CountryName",

cf.gdp AS "GDP"

FROM

country\_fact cf

JOIN

region r ON cf.cid = r.cid

WHERE

r.continent6 = 'America';

Result:

"RankNo" "RegionName" "CountryName" "GDP"

1 "Central America and the Caribbean" "Guatemala" 33000000000

2 "Central America and the Caribbean" "Puerto Rico" 26800000000

3 "Central America and the Caribbean" "Dominican Republic" 24000000000

4 "Central America and the Caribbean" "Costa Rica" 16900000000

5 "Central America and the Caribbean" "Trinidad and Tobago" 15000000000

6 "Central America and the Caribbean" "Cuba" 14000000000

7 "Central America and the Caribbean" "Panama" 12300000000

8 "Central America and the Caribbean" "El Salvador" 9800000000

9 "Central America and the Caribbean" "Honduras" 9700000000

10 "Central America and the Caribbean" "Jamaica" 7800000000

11 "Central America and the Caribbean" "Nicaragua" 6400000000

12 "Central America and the Caribbean" "Haiti" 5600000000

13 "Central America and the Caribbean" "The Bahamas" 4400000000

14 "Central America and the Caribbean" "Martinique" 3900000000

15 "Central America and the Caribbean" "Guadeloupe" 3800000000

16 "Central America and the Caribbean" "Barbados" 2400000000

17 "Central America and the Caribbean" "Netherlands Antilles" 1850000000

18 "Central America and the Caribbean" "Virgin Islands" 1200000000

19 "Central America and the Caribbean" "Aruba" 1100000000

20 "Central America and the Caribbean" "Cayman Islands" 700000000

21 "Central America and the Caribbean" "Saint Lucia" 610000000

22 "Central America and the Caribbean" "Belize" 575000000

23 "Central America and the Caribbean" "Antigua and Barbuda" 400000000

24 "Central America and the Caribbean" "Grenada" 258000000

25 "Central America and the Caribbean" "Saint Vincent and the Grenadines" 235000000

26 "Central America and the Caribbean" "Saint Kitts and Nevis" 210000000

27 "Central America and the Caribbean" "Dominica" 200000000

28 "Central America and the Caribbean" "British Virgin Islands" 133000000

29 "Central America and the Caribbean" "Turks and Caicos Islands" 80800000

30 "Central America and the Caribbean" "Montserrat" 55600000

31 "Central America and the Caribbean" "Anguilla" 49000000

32 "Central America and the Caribbean" "Navassa Island" 0

1 "North America" "United States" 6738400000000

2 "North America" "Mexico" 728700000000

3 "North America" "Canada" 639800000000

4 "North America" "Bermuda" 1700000000

5 "North America" "Saint Pierre and Miquelon" 66000000

1 "South America" "Brazil" 886300000000

2 "South America" "Argentina" 270800000000

3 "South America" "Venezuela" 178300000000

4 "South America" "Colombia" 172400000000

5 "South America" "Chile" 97700000000

6 "South America" "Peru" 73600000000

7 "South America" "Ecuador" 41100000000

8 "South America" "Uruguay" 23000000000

9 "South America" "Bolivia" 18300000000

10 "South America" "Paraguay" 15400000000

11 "South America" "Guyana" 1400000000

12 "South America" "Suriname" 1200000000

13 "South America" "French Guiana" 800000000

14 "South America" "Falkland Islands (Islas Malvinas)" 0

**7.** Explain on the distinctions between the SQL query statements for Q5 and Q6, focusing on the differences in their approach to ranking countries by GDP within America.

Result:

**Distinctions:**

1. **Partitioning Approach:**
   * **Q5:** The ROW\_NUMBER() function is used with PARTITION BY r.region\_name, which means that the ranking is done separately within each region. Each region's countries are ranked independently based on their GDP, and the ranking restarts for each new region.
   * **Q6:** There's no explicit partitioning in the ROW\_NUMBER() function. All countries within 'America' are considered together when assigning ranks based on GDP. There's a single sequence of ranks across all regions within 'America', and countries from different regions may have the same rank.
2. **Ranking Behavior:**
   * **Q5:** The rank assigned to a country is specific to its region. For example, if Country A is ranked 1st in Region X, it doesn't affect the ranking of Country A in Region Y. Each region has its own ranking sequence based on GDP.
   * **Q6:** The rank assigned to a country is not specific to its region. It's a global ranking across all regions within 'America'. For example, if Country A is ranked 1st, it means it has the highest GDP among all countries within 'America', regardless of its region.

In summary, the distinction between the two queries lies in how they handle the partitioning of data and the assignment of ranks within 'America'. Q5 assigns ranks within each region independently, while Q6 assigns a single global rank across all regions within 'America'.

**8.** Construct an SQL query to find the country with the highest GDP in each region of America, ensuring each identified country has the highest rank (Rank 1) in its region. Adjust the SQL statement from Q6 for this purpose.

Query:

SELECT

ROW\_NUMBER() OVER (PARTITION BY r.region\_name ORDER BY cf.gdp DESC) AS "RankNo",

r.region\_name AS "RegionName",

r.country\_name AS "CountryName",

cf.gdp AS "GDP"

FROM

country\_fact cf

JOIN

region r ON cf.cid = r.cid

WHERE

(r.region\_name, cf.gdp) IN

(SELECT

r.region\_name,

MAX(cf.gdp) AS max\_gdp

FROM

country\_fact cf

JOIN

region r ON cf.cid = r.cid

WHERE

r.continent6 = 'America'

GROUP BY

r.region\_name);

Result:

A screenshot of a computer

Description automatically generated

**9.** Create an SQL query to find the country with the lowest GDP in each region of America by adapting the approach from Q8.

Query:

SELECT

SELECT

ROW\_NUMBER() OVER (PARTITION BY r.region\_name ORDER BY cf.gdp ASC) AS "RankNo",

r.region\_name AS "RegionName",

r.country\_name AS "CountryName",

cf.gdp AS "GDP"

FROM

country\_fact cf

JOIN

region r ON cf.cid = r.cid

WHERE

(r.region\_name, cf.gdp) IN

(SELECT

r.region\_name,

MIN(cf.gdp) AS min\_gdp

FROM

country\_fact cf

JOIN

region r ON cf.cid = r.cid

WHERE

r.continent6 = 'America'

GROUP BY

r.region\_name);

Result:

A screenshot of a computer

Description automatically generated

**10.** Use the CUBE function to write a single SQL statement that counts the number of countries by REGION (i.e., continent4 region) and POPULATION dimensions. Ensure your query result matches the provided example, including column headings. A CASE WHEN clause may be necessary for displaying “SUB TOTAL” value as a derived value.

Query:

SELECT

CASE

WHEN region\_name IS NULL THEN 'SUB TOTAL'

ELSE region\_name

END AS "RegionName",

CASE

WHEN range IS NULL THEN 'SUB TOTAL'

ELSE range

END AS "PopulationRange",

COUNT(country\_name) AS "Number of Countries"

FROM

region

JOIN

country\_fact ON region.cid = country\_fact.cid

JOIN

population ON country\_fact.pid = population.pid

GROUP BY

CUBE(region\_name, range)

ORDER BY

region\_name, range;

Result:

"RegionName" "PopulationRange" "Number of Countries"

"Africa" " < 72751" 7

"Africa" " > 10647511" 17

"Africa" " 10647511 - 2826683" 20

"Africa" " 2826683 - 72751" 15

"Africa" "SUB TOTAL" 59

"Antarctic Region" " < 72751" 5

"Antarctic Region" "SUB TOTAL" 5

"Arctic Region" " < 72751" 4

"Arctic Region" " 2826683 - 72751" 1

"Arctic Region" "SUB TOTAL" 5

"Asia" " > 10647511" 11

"Asia" " 2826683 - 72751" 3

"Asia" "SUB TOTAL" 14

"Central America and the Caribbean" " < 72751" 9

"Central America and the Caribbean" " > 10647511" 2

"Central America and the Caribbean" " 10647511 - 2826683" 7

"Central America and the Caribbean" " 2826683 - 72751" 14

"Central America and the Caribbean" "SUB TOTAL" 32

"Commonwealth of Independent States - Central Asian States" " > 10647511" 2

"Commonwealth of Independent States - Central Asian States" " 10647511 - 2826683" 3

"Commonwealth of Independent States - Central Asian States" "SUB TOTAL" 5

"Commonwealth of Independent States - European States" " > 10647511" 1

"Commonwealth of Independent States - European States" " 10647511 - 2826683" 4

"Commonwealth of Independent States - European States" "SUB TOTAL" 5

"Ethnic Groups in Eastern Europe, Europe" " > 10647511" 2

"Ethnic Groups in Eastern Europe, Europe" " 10647511 - 2826683" 8

"Ethnic Groups in Eastern Europe, Europe" " 2826683 - 72751" 2

"Ethnic Groups in Eastern Europe, Europe" "SUB TOTAL" 12

"Europe" " < 72751" 8

"Europe" " > 10647511" 6

"Europe" " 10647511 - 2826683" 11

"Europe" " 2826683 - 72751" 6

"Europe" "SUB TOTAL" 31

"Middle East" " > 10647511" 6

"Middle East" " 10647511 - 2826683" 5

"Middle East" " 2826683 - 72751" 7

"Middle East" "SUB TOTAL" 18

"North America" " < 72751" 2

"North America" " > 10647511" 3

"North America" "SUB TOTAL" 5

"Oceania" " < 72751" 21

"Oceania" " > 10647511" 1

"Oceania" " 10647511 - 2826683" 2

"Oceania" " 2826683 - 72751" 10

"Oceania" "SUB TOTAL" 34

"South America" " < 72751" 1

"South America" " > 10647511" 7

"South America" " 10647511 - 2826683" 3

"South America" " 2826683 - 72751" 3

"South America" "SUB TOTAL" 14

"Southeast Asia" " < 72751" 5

"Southeast Asia" " > 10647511" 7

"Southeast Asia" " 10647511 - 2826683" 4

"Southeast Asia" " 2826683 - 72751" 2

"Southeast Asia" "SUB TOTAL" 18

"World" " < 72751" 5

"World" " 2826683 - 72751" 3

"World" "SUB TOTAL" 8

"SUB TOTAL" " < 72751" 67

"SUB TOTAL" " > 10647511" 65

"SUB TOTAL" " 10647511 - 2826683" 67

"SUB TOTAL" " 2826683 - 72751" 66

"SUB TOTAL" "SUB TOTAL" 265

A screenshot of a computer

Description automatically generated

**11.** To refine the results from Q10 by excluding subtotals by country and the overall total, write an SQL query employing the ROLLUP function to achieve the desired output.

Query:

SELECT

CASE

WHEN region\_name IS NULL THEN 'SUB TOTAL'

ELSE region\_name

END AS "RegionName",

CASE

WHEN range IS NULL THEN 'SUB TOTAL'

ELSE range

END AS "PopulationRange",

COUNT(country\_name) AS "Number of Countries"

FROM

region

JOIN

country\_fact ON region.cid = country\_fact.cid

JOIN

population ON country\_fact.pid = population.pid

GROUP BY

ROLLUP(region\_name, range)

HAVING

region\_name IS NOT NULL AND range IS NOT NULL

ORDER BY

region\_name, range;

Result:

"RegionName" "PopulationRange" "Number of Countries"

"Africa" " < 72751" 7

"Africa" " > 10647511" 17

"Africa" " 10647511 - 2826683" 20

"Africa" " 2826683 - 72751" 15

"Antarctic Region" " < 72751" 5

"Arctic Region" " < 72751" 4

"Arctic Region" " 2826683 - 72751" 1

"Asia" " > 10647511" 11

"Asia" " 2826683 - 72751" 3

"Central America and the Caribbean" " < 72751" 9

"Central America and the Caribbean" " > 10647511" 2

"Central America and the Caribbean" " 10647511 - 2826683" 7

"Central America and the Caribbean" " 2826683 - 72751" 14

"Commonwealth of Independent States - Central Asian States" " > 10647511" 2

"Commonwealth of Independent States - Central Asian States" " 10647511 - 2826683" 3

"Commonwealth of Independent States - European States" " > 10647511" 1

"Commonwealth of Independent States - European States" " 10647511 - 2826683" 4

"Ethnic Groups in Eastern Europe, Europe" " > 10647511" 2

"Ethnic Groups in Eastern Europe, Europe" " 10647511 - 2826683" 8

"Ethnic Groups in Eastern Europe, Europe" " 2826683 - 72751" 2

"Europe" " < 72751" 8

"Europe" " > 10647511" 6

"Europe" " 10647511 - 2826683" 11

"Europe" " 2826683 - 72751" 6

"Middle East" " > 10647511" 6

"Middle East" " 10647511 - 2826683" 5

"Middle East" " 2826683 - 72751" 7

"North America" " < 72751" 2

"North America" " > 10647511" 3

"Oceania" " < 72751" 21

"Oceania" " > 10647511" 1

"Oceania" " 10647511 - 2826683" 2

"Oceania" " 2826683 - 72751" 10

"South America" " < 72751" 1

"South America" " > 10647511" 7

"South America" " 10647511 - 2826683" 3

"South America" " 2826683 - 72751" 3

"Southeast Asia" " < 72751" 5

"Southeast Asia" " > 10647511" 7

"Southeast Asia" " 10647511 - 2826683" 4

"Southeast Asia" " 2826683 - 72751" 2

"World" " < 72751" 5

"World" " 2826683 - 72751" 3

A screenshot of a computer

Description automatically generated

**12.** Implement an SQL statement to display the result of Q11 in a cross-tabulation format. Your query should replicate the provided example result. **Hint**: Consider utilizing the PIVOT function for this task.

Query:

SELECT

CASE

WHEN "PopulationRange" IS NULL THEN 'SUB TOTAL'

ELSE "PopulationRange"

END AS "PopulationRange",

SUM("North\_America") AS "North\_America",

SUM("Central\_America\_and\_the\_Caribbean") AS "Central\_America\_and\_the\_Caribbean",

SUM("South\_America") AS "South\_America"

FROM

(SELECT

CASE

WHEN range IS NULL THEN 'SUB TOTAL'

ELSE range

END AS "PopulationRange",

region\_name AS "RegionName",

COUNT(country\_name) AS "Number\_of\_Countries",

CASE

WHEN region\_name = 'North America' THEN COUNT(country\_name)

ELSE 0

END AS "North\_America",

CASE

WHEN region\_name = 'Central America and the Caribbean' THEN COUNT(country\_name)

ELSE 0

END AS "Central\_America\_and\_the\_Caribbean",

CASE

WHEN region\_name = 'South America' THEN COUNT(country\_name)

ELSE 0

END AS "South\_America"

FROM

region

JOIN

country\_fact ON region.cid = country\_fact.cid

JOIN

population ON country\_fact.pid = population.pid

GROUP BY

GROUPING SETS ((region\_name, range), (range))

HAVING

region\_name IS NOT NULL OR range IS NOT NULL) AS SourceTable

GROUP BY

"PopulationRange"

ORDER BY

"PopulationRange";

Result:

A screenshot of a computer

Description automatically generated