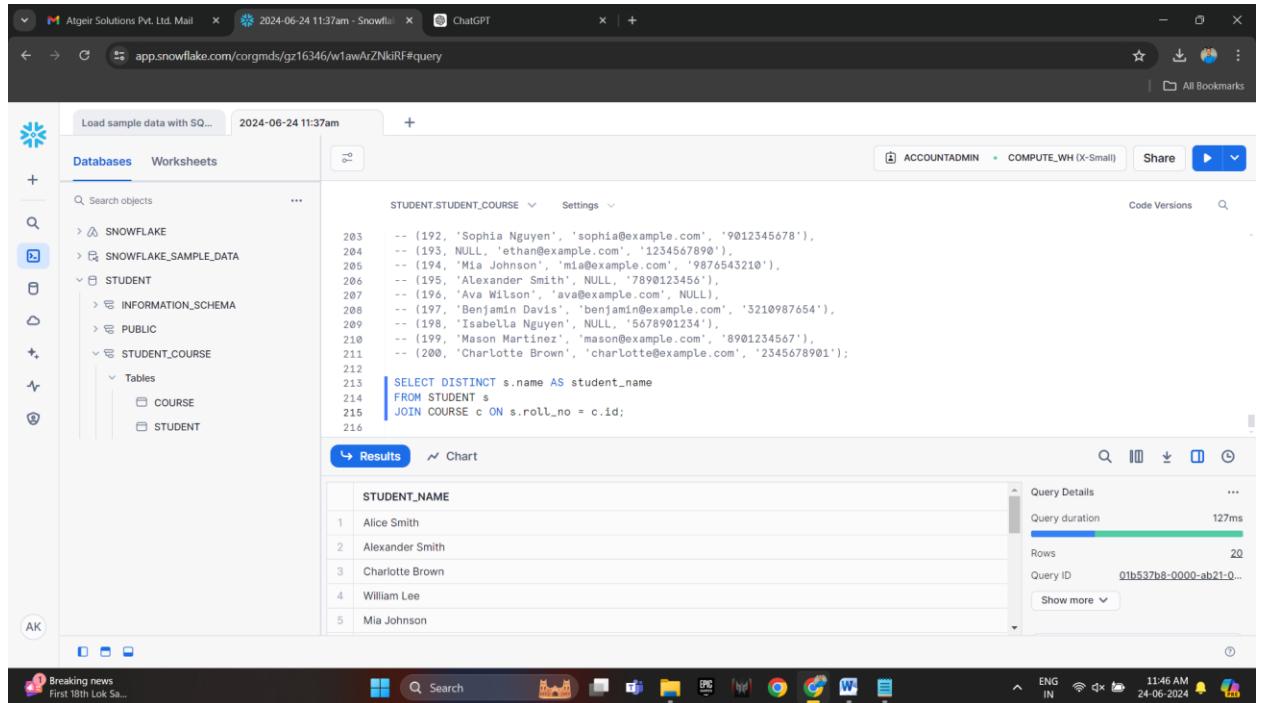


SQL Assignment

college_db.course

1. Retrieve the names of students who are enrolled in any course.



The screenshot shows the Snowflake web interface. On the left, the sidebar lists databases (SNOWFLAKE, SNOWFLAKE_SAMPLE_DATA), a STUDENT schema containing INFORMATION_SCHEMA, PUBLIC, and STUDENT,COURSE tables, and a COURSE table under STUDENT,COURSE. The main area displays a query in the editor:

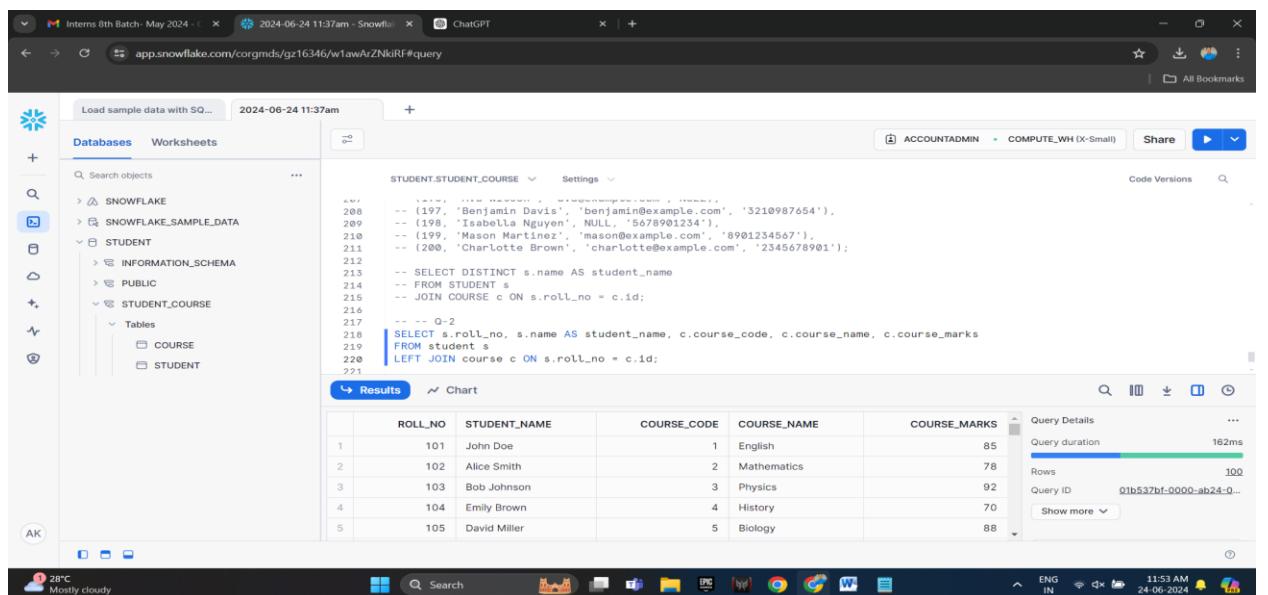
```
203 -- (192, 'Sophia Nguyen', 'sophia@example.com', '9012345678'),
204 -- (193, NULL, 'ethan@example.com', '1234567890'),
205 -- (194, 'Mia Johnson', 'mia@example.com', '9876543210'),
206 -- (195, 'Alexander Smith', NULL, '7890123456'),
207 -- (196, 'Ava Wilson', 'ava@example.com', NULL),
208 -- (197, 'Benjamin Davis', 'benjamin@example.com', '3210987654'),
209 -- (198, 'Isabella Nguyen', NULL, '5678901234'),
210 -- (199, 'Mason Martinez', 'mason@example.com', '8901234567'),
211 -- (200, 'Charlotte Brown', 'charlotte@example.com', '2345678901');
212
213 SELECT DISTINCT s.name AS student_name
214 FROM STUDENT s
215 JOIN COURSE c ON s.roll_no = c.id;
216
```

The results table shows the following data:

STUDENT_NAME
Alice Smith
Alexander Smith
Charlotte Brown
William Lee
Mia Johnson

Query Details: Query duration 127ms, Rows 20, Query ID 01b537b8-0000-ab21-0...

2. List all students along with the courses they are enrolled in, if any.



The screenshot shows the Snowflake web interface. The sidebar is identical to the previous screenshot. The main area displays a more complex query in the editor:

```
208 -- (197, 'Benjamin Davis', 'benjamin@example.com', '3210987654'),
209 -- (198, 'Isabella Nguyen', NULL, '5678901234'),
210 -- (199, 'Mason Martinez', 'mason@example.com', '8901234567'),
211 -- (200, 'Charlotte Brown', 'charlotte@example.com', '2345678901');
212
213 -- SELECT DISTINCT s.name AS student_name
214 -- FROM STUDENT s
215 -- JOIN COURSE c ON s.roll_no = c.id;
216
217 -- -- Q-2
218 -- SELECT s.roll_no, s.name AS student_name, c.course_code, c.course_name, c.course_marks
219 -- FROM student s
220 -- LEFT JOIN course c ON s.roll_no = c.id;
221
```

The results table shows the following data:

ROLL_NO	STUDENT_NAME	COURSE_CODE	COURSE_NAME	COURSE_MARKS
1	101	1	English	85
2	102	2	Mathematics	78
3	103	3	Physics	92
4	104	4	History	70
5	105	5	Biology	88

Query Details: Query duration 162ms, Rows 100, Query ID 01b537bf-0000-ab24-0...

3. Show the course codes and names along with the corresponding student names for all students who are enrolled in any course.

```

214 -- FROM STUDENT s
215 -- JOIN COURSE c ON s.roll_no = c.id;
216
217 -- -- Q-2
218 -- SELECT s.roll_no, s.name AS student_name,
219 -- CASE WHEN c.course_code IS NOT NULL THEN c.course_code ELSE NULL END AS course_code,
220 -- c.course_name, c.course_marks
221 -- FROM STUDENT s
222 -- LEFT JOIN COURSE c ON s.roll_no = c.id;
223
224 -- Q-3
225 -- SELECT c.course_code, c.course_name, s.name AS student_name
226 -- FROM COURSE c
227 JOIN STUDENT s ON c.id = s.roll_no;

```

	COURSE_CODE	COURSE_NAME	STUDENT_NAME
1	1	English	Alice Smith
2	2	Mathematics	Bob Johnson
3	3	Physics	Emily Brown
4	4	History	David Miller
5	5	Biology	

4. Retrieve the details of all courses and their corresponding students, including courses with no enrolled students.

```

217 -- -- Q-2
218 -- SELECT s.roll_no, s.name AS student_name, c.course_code, c.course_name, c.course_marks
219 -- FROM STUDENT s
220 -- LEFT JOIN COURSE c ON s.roll_no = c.id;
221
222 -- -- Q-3
223 -- -- SELECT c.course_code, c.course_name, s.name AS student_name
224 -- -- FROM COURSE c
225 -- -- JOIN STUDENT s ON c.id = s.roll_no;
226
227 -- -- Q-4
228 -- SELECT c.course_code, c.course_name, COALESCE(s.name, 'No student enrolled') AS student_name
229 -- FROM COURSE c
230 LEFT JOIN STUDENT s ON c.id = s.roll_no;

```

	COURSE_CODE	COURSE_NAME	STUDENT_NAME
1	1	English	John Doe
2	2	Mathematics	Alice Smith
3	3	Physics	Bob Johnson
4	4	History	Emily Brown
5	5	Biology	No student enrolled

5. Display the details of students and the courses they are enrolled in.

The screenshot shows the Snowflake web interface. On the left sidebar, under 'Databases', 'SNOWFLAKE' is selected. Under 'STUDENT' > 'Tables', 'STUDENT_COURSE' is selected. The main area displays a query result table titled 'STUDENT.STUDENT_COURSE'. The table has columns: ROLL_NO, STUDENT_NAME, COURSE_CODE, COURSE_NAME, and COURSE_MARKS. The data is as follows:

ROLL_NO	STUDENT_NAME	COURSE_CODE	COURSE_NAME	COURSE_MARKS
1	John Doe	1	English	85
2	Alice Smith	2	Mathematics	78
3	Bob Johnson	3	Physics	92
4	Emily Brown	4	History	70
5	David Miller	5	Biology	88

On the right side, there is a 'Query Details' panel showing: Query duration 25ms, Rows 100, and Query ID 01b537ca-0000-ab24-0... . Below the table, there are buttons for 'Results' and 'Chart'.

6. Retrieve the details of all students and their corresponding courses, including students who are not enrolled in any course and courses with no enrolled students.

The screenshot shows the Snowflake web interface with a modified query. The main area displays a query result table titled 'STUDENT.STUDENT_COURSE'. The table has columns: ROLL_NO, STUDENT_NAME, COURSE_CODE, COURSE_NAME, and COURSE_MARKS. The data is identical to the previous screenshot:

ROLL_NO	STUDENT_NAME	COURSE_CODE	COURSE_NAME	COURSE_MARKS
1	John Doe	1	English	85
2	Alice Smith	2	Mathematics	78
3	Bob Johnson	3	Physics	92
4	Emily Brown	4	History	70
5	David Miller	5	Biology	88

On the right side, there is a 'Query Details' panel showing: Query duration 27ms, Rows 100, and Query ID 01b537cd-0000-ab24-0... . Below the table, there are buttons for 'Results' and 'Chart'.

7. Show all courses and their corresponding student names, if any student is enrolled.

```

237 -- Q-6
238 -- SELECT s.roll_no, s.name AS student_name, c.course_code, c.course_name, c.course_marks
239 -- FROM student s
240 -- LEFT JOIN course c ON s.roll_no = c.id
241 -- UNION ALL
242 -- SELECT NULL AS roll_no, NULL AS student_name, c.course_code, c.course_name, c.course_marks
243 -- FROM course c
244 -- LEFT JOIN student s ON s.roll_no = c.id
245 -- WHERE s.roll_no IS NULL;
246
247 -- Q-7
248 -- SELECT c.course_code, c.course_name, COALESCE(s.name, 'No student enrolled') AS student_name
249 -- FROM course c
250 -- LEFT JOIN student s ON c.id = s.roll_no;

```

	COURSE_CODE	COURSE_NAME	STUDENT_NAME
1	1	English	John Doe
2	2	Mathematics	Alice Smith
3	3	Physics	Bob Johnson
4	4	History	Emily Brown
5	5	Biology	David Miller

8. Retrieve the details of all students and their corresponding courses, including students who are not enrolled in any course.

```

242 -- SELECT NULL AS roll_no, NULL AS student_name, c.course_code, c.course_name, c.course_marks
243 -- FROM course c
244 -- LEFT JOIN student s ON s.roll_no = c.id
245 -- WHERE s.roll_no IS NULL;
246
247 -- Q-7
248 -- SELECT c.course_code, c.course_name, COALESCE(s.name, 'No student enrolled') AS student_name
249 -- FROM course c
250 -- LEFT JOIN student s ON c.id = s.roll_no;
251
252 -- Q-8
253 -- SELECT s.roll_no, s.name AS student_name, COALESCE(c.course_code, -1) AS course_code, c.course_name, c.course_marks
254 -- FROM student s
255 -- LEFT JOIN course c ON s.roll_no = c.id;

```

	ROLL_NO	STUDENT_NAME	COURSE_CODE	COURSE_NAME	COURSE_MARKS
1	101	John Doe	1	English	85
2	102	Alice Smith	2	Mathematics	78
3	103	Bob Johnson	3	Physics	92
4	104	Emily Brown	4	History	70
5	105	David Miller	5	Biology	88

9. List all courses along with the names of students enrolled in each course, if any.

```

249 -- FROM course c
250 -- LEFT JOIN student s ON c.id = s.roll_no;
251
252 -- Q-8
253 -- SELECT s.roll_no, s.name AS student_name, COALESCE(c.course_code, -1) AS course_code, c.course_name, c.course_marks
254 -- FROM student s
255 -- LEFT JOIN course c ON s.roll_no = c.id;
256
257 -- Q-9
258 -- SELECT c.course_code, c.course_name,
259 -- LISTAGG(s.name, ', ') WITHIN GROUP (ORDER BY s.name) AS student_names
260 -- FROM course c
261 -- LEFT JOIN student s ON c.id = s.roll_no
262 -- GROUP BY c.course_code, c.course_name;

```

	COURSE_CODE	COURSE_NAME	STUDENT_NAMES
1	3	Physics	Bob Johnson, Bob Johnson, Bob Johnson, Bob Johnson, Bob Johnson, Emma Martinez
2	5	Biology	Alexander Smith, Alexander Smith, Alexander Smith, Alexander Smith, Alexander Smith
3	2	Mathematics	Alice Smith, Alice Smith, Alice Smith, Alice Smith, Benjamin Davis, Benjamin Davis
4	1	English	Ava Wilson, Ava Wilson, Ava Wilson, Ava Wilson, Ava Wilson, John Doe, John Doe, John Doe
5	4	History	Emily Brown, Emily Brown, Emily Brown, Emily Brown, Emily Brown, James Garcia, James Garcia

Query Details
Query duration 126ms
Rows 5
Query ID 01b537d1-0000-ab21-0...
Show more

10. Show all courses and their corresponding student names, including courses with no enrolled students.

```

254 -- FROM student s
255 -- LEFT JOIN course c ON s.roll_no = c.id;
256
257 -- Q-9
258 -- SELECT c.course_code, c.course_name,
259 -- LISTAGG(s.name, ', ') WITHIN GROUP (ORDER BY s.name) AS student_names
260 -- FROM course c
261 -- LEFT JOIN student s ON c.id = s.roll_no
262 -- GROUP BY c.course_code, c.course_name;
263
264 -- Q-10
265 -- SELECT c.course_code, c.course_name, COALESCE(s.name, 'No student enrolled') AS student_name
266 -- FROM course c
267 -- LEFT JOIN student s ON c.id = s.roll_no;

```

	COURSE_CODE	COURSE_NAME	STUDENT_NAME
1	1	English	John Doe
2	2	Mathematics	Alice Smith
3	3	Physics	Bob Johnson
4	4	History	Emily Brown
5	5	Biology	David Miller

Query Details
Query duration 16ms
Rows 100
Query ID 01b537d2-0000-ab21-0...
Show more

11. Show all students and their corresponding course details, if any student is enrolled.

Load sample data with SQL... 2024-06-24 11:37am +

Databases Worksheets

Search objects ...

Snowflake SNOWFLAKE SNOWFLAKE_SAMPLE_DATA STUDENT STUDENT COURSE STUDENT COURSE

```

STUDENT.STUDENT_COURSE Settings
259 -- LISTAGG(s.name, ', ') WITHIN GROUP (ORDER BY s.name) AS student_names
260 -- FROM course c
261 -- LEFT JOIN student s ON c.id = s.roll_no
262 -- GROUP BY c.course_code, c.course_name;
263
264 -- Q-10
265 -- SELECT c.course_code, c.course_name, COALESCE(s.name, 'No student enrolled') AS student_name
266 -- FROM course c
267 -- LEFT JOIN student s ON c.id = s.roll_no;
268
269 -- Q-11
270 -- SELECT s.roll_no, s.name AS student_name, c.course_code, c.course_name, c.course_marks
271 FROM student s
272 LEFT JOIN course c ON s.roll_no = c.id;

```

Results Chart

ROLL_NO	STUDENT_NAME	COURSE_CODE	COURSE_NAME	COURSE_MARKS
1	John Doe	1	English	85
2	Alice Smith	2	Mathematics	78
3	Bob Johnson	3	Physics	92
4	Emily Brown	4	History	70
5	David Miller	5	Biology	88

Query Details

- Query duration 18ms
- Rows 100
- Query ID 01b537d4-0000-ab24-...
- Show more

ENG IN 12:14 PM 24-06-2024

12. List all students along with the courses they are enrolled in, if any, and also include students who are not enrolled in any course.

Load sample data with SQL... 2024-06-24 11:37am +

Databases Worksheets

Search objects ...

Snowflake SNOWFLAKE SNOWFLAKE_SAMPLE_DATA STUDENT STUDENT COURSE STUDENT COURSE

```

STUDENT.STUDENT_COURSE Settings
279
280 FROM STUDENT s
281 LEFT JOIN COURSE c ON s.roll_no = c.id
282 UNION ALL
283 SELECT s.roll_no, s.name AS student_name,
284 -1 AS course_code,
285 NULL AS course_name,
286 NULL AS course_marks
287 FROM STUDENT s
288 WHERE s.roll_no NOT IN (SELECT id FROM COURSE);
289

```

Results Chart

ROLL_NO	STUDENT_NAME	COURSE_CODE	COURSE_NAME	COURSE_MARKS
1	John Doe	1	English	85
2	Alice Smith	2	Mathematics	78
3	Bob Johnson	3	Physics	92
4	Emily Brown	4	History	70
5	David Miller	5	Biology	88
6	Sarah Wilson	1	English	null
7	Michael Davis	2	Mathematics	82
8	Emma Martinez	3	Physics	90

Query Details

- Query duration 212ms
- Rows 100
- Query ID 01b537d4-0000-ab21-...
- Show more

ROLL_NO #

ENG IN 12:21 PM 24-06-2024

world.city

1. Retrieve the top 5 countries with the highest ratio of capital city population to country population.

```
SELECT c.Name AS country_name,
       ct.Name AS capital_city,
       c.Population AS country_population,
       ct.Population AS capital_population,
       (ct.Population / c.Population) AS capital_to_country_population_ratio
  FROM country c
 JOIN city ct ON c.Capital = ct.ID
 ORDER BY capital_to_country_population_ratio DESC
 LIMIT 5;
```

COUNTRY_NAME	CAPITAL_CITY	COUNTRY_POPULATION	CAPITAL_POPULATION	CAPITAL_TO_COUNTRY_RATIO
Singapore	Singapore	3567000	4017733	
Gibraltar	Gibraltar	25000	27025	
Macao	Macao	473000	437500	
Pitcairn	Adamstown	50	42	
Saint Pierre and Miquelon	Saint-Pierre	7000	5808	

2. List the continents where the average life expectancy is above the global average, along with the number of countries in each continent.

```
-- FROM country c
-- JOIN city ct ON c.Capital = ct.ID
-- ORDER BY capital_to_country_population_ratio DESC
-- LIMIT 5;
SELECT c.Continent,
       AVG(c.LifeExpectancy) AS avg_life_expectancy,
       COUNT(*) AS num_countries
  FROM country c
 WHERE c.LifeExpectancy > (SELECT AVG(LifeExpectancy) FROM country)
 GROUP BY c.Continent;
```

CONTINENT	AVG_LIFE_EXPECTANCY	NUM_COUNTRIES
North America	74.1411765	34
Europe	75.6190476	42
Asia	72.8387097	31
South America	73.1700000	10
Oceania	72.7285714	14

3. Find the country with the highest GNP per capita (GNP divided by population) and its capital city.

```

SELECT c.Name AS country_name,
       ct.Name AS capital_city,
       (c.GNP / c.Population) AS GNP_per_capita
FROM country c
JOIN city ct ON c.Capital = ct.ID
ORDER BY GNP_per_capita DESC
LIMIT 1;
  
```

COUNTRY_NAME	CAPITAL_CITY	GNP_PER_CAPITA
Luxembourg	Luxembourg [Luxemburg/Lëtzebuerg]	0.03745926

Query Details

- Query duration: 235ms
- Rows: 1
- Query ID: 01b537fe-0000-ab24-0...

4. Retrieve the top 3 languages spoken in each continent, along with the percentage of speakers and the number of countries where the language is official.

```

WITH RankedLanguages AS (
  SELECT cl.Language,
         c.Continent,
         ROW_NUMBER() OVER (PARTITION BY c.Continent ORDER BY cl.Percentage DESC) AS LanguageRank,
         cl.Percentage,
         COUNT(*) OVER (PARTITION BY c.Continent, cl.Language) AS NumCountriesOfficial
  FROM countrylanguage cl
  JOIN country c ON cl.CountryCode = c.Code
  WHERE cl.Official = 'T'
)
SELECT Continent,
       Language,
       Percentage,
       NumCountriesOfficial
  
```

CONTINENT	LANGUAGE	PERCENTAGE	NUMCOUNTRIESOFFICIAL
Africa	Rwanda	100.0	1
Africa	Arabic	100.0	10
Africa	Malagasy	98.9	1
Oceania	Kiribati	98.9	1

Query Details

- Query duration: 278ms
- Rows: 18
- Query ID: 01b537fe-0000-ab24-0...

5. List the countries where the head of state is a woman and their respective capital cities.

```

45 -- WHLkt. LanguageRank <= 5;
46
47 -- Q-5
48 SELECT c.Name AS CountryName,
49     ci.Name AS CapitalCity
50 FROM country c
51 JOIN city ci ON c.Capital = ci.ID
52 WHERE c.HeadOfState LIKE '%woman%';
53
54 -- Q-6
55 -- WITH OfficialLanguageSpeakers AS (
56 --     SELECT cl.CountryCode,
57 --         cl.Language,
58 --         cl.Percentage AS OfficialLanguagePercentage
59 --     FROM countrylanguage cl
60 --     WHERE cl.Official = 'T'
61 ),

```

Results

COUNTRYNAME	CAPITALCITY
Query produced no results	

Query Details

- Query duration: 119ms
- Rows: 0
- Query ID: 01b537ff-0000-ab24-0...

- Find the countries where more than 50% of the population speaks an official language, along with the percentage of population speaking the official language.

```

63     SELECT c.Code,
64         c.Population
65     FROM country c
66 )
67     SELECT c.Name AS CountryName,
68         ci.Name AS CapitalCity,
69         ol.Language AS OfficialLanguage,
70         ol.OfficialLanguagePercentage AS PercentageSpeakingOfficialLanguage
71     FROM country c
72     JOIN city ci ON c.Capital = ci.ID
73     JOIN OfficialLanguageSpeakers ol ON c.Code = ol.CountryCode
74     JOIN CountryPopulation cp ON c.Code = cp.Code
75     WHERE ol.OfficialLanguagePercentage > 50;
76

```

Results

COUNTRYNAME	CAPITALCITY	OFFICIALLANGUAGE	PERCENTAGESPEAKINGOFFICIALLANGUAGE
Afghanistan	Kabul	Pashto	52.4
Netherlands	Amsterdam	Dutch	95.6
Netherlands Antilles	Willemstad	Papiamento	86.2
Albania	Tirana	Albanian	97.9
Algeria	Alger	Arabic	86.0

Query Details

- Query duration: 247ms
- Rows: 130
- Query ID: 01b53800-0000-ab24-0...

- Identify the continent with the highest total surface area covered by countries whose names start with the letter 'A'.

```

74 -- JOIN CountryPopulation cp ON c.Code = cp.Code
75 -- WHERE ol.OfficialLanguagePercentage > 50;
76
77 -- Q-7
78 SELECT Continent,
79      SUM(SurfaceArea) AS TotalSurfaceArea
80 FROM country
81 WHERE Name LIKE '%'
82 GROUP BY Continent
83 ORDER BY TotalSurfaceArea DESC
84 LIMIT 1;

```

CONTINENT	TOTALSURFACEAREA
Antarctica	13120000.00

Query Details
Query duration 45ms
Rows 1
Query ID 01b53801-0000-ab24-0...
Show more

8. List the cities where the population is higher than the capital city of their respective country.

```

86 -- Q-8
87 SELECT c1.Name AS CityName,
88        c2.Name AS CapitalCityName,
89        c1.Population AS CityPopulation,
90        c2.Population AS CapitalPopulation
91 FROM city c1
92 JOIN country c2 ON c1.CountryCode = c2.Code
93 WHERE c1.Population > c2.Population;
94
95 -- Q-9
96 -- WITH AveragePopulationByContinent AS (

```

CITYNAME	CAPITALCITYNAME	CITYPOPULATION	CAPITALPOPULATION
Gibraltar	Gibraltar	27025	25000
Singapore	Singapore	4017733	3567000

Query Details
Query duration 147ms
Rows 2
Query ID 01b53801-0000-ab21-0...
Show more

9. Find the countries with a population greater than the average population of all countries in their continent, along with their populations and continents.

```

SELECT c.Name AS CountryName,
       c.Population AS CountryPopulation,
       c.Continent
  FROM country c
 JOIN AveragePopulationByContinent ap ON c.Continent = ap.Continent
 WHERE c.Population > ap.AvgPopulation;

```

COUNTRYNAME	COUNTRYPOPULATION	CONTINENT
Argentina	37032000	South America
Australia	18886000	Oceania
Bangladesh	129155000	Asia
Brazil	170115000	South America
Canada	31147000	North America
China	1277558000	Asia
Côte d'Ivoire	14786000	Africa

10. Retrieve the countries where the difference between the current population and the population at independence year is greater than 10 million, along with their current populations and independence years.

```

SELECT c.Name AS CountryName,
       c.Population AS CurrentPopulation,
       c.IndependenceYear AS IndependenceYear,
       c.Population - ci.Population AS PopulationDifference
  FROM country c
 JOIN country ci ON c.Name = ci.Name
 WHERE c.IndependenceYear IS NOT NULL
   AND c.Population - ci.Population > 10000000;

```

COUNTRYNAME	CURRENTPOPULATION	INDEPENDENCEYEAR	POPULATIONDIFFERENCE
Query produced no results			