

COMP 306: Database Management Systems

Spring 2023 - Homework #3

Use the 'middle.js,' 'server.js,' 'q2.html', and 'q4.html' files to solve the questions below. In all questions, we will use the same dataset that we used in the previous homework (named world.sql). If you changed the dataset while doing the previous homework, please delete your copy first and re-populate it from scratch using the version we are providing in this HW.

Question 1. [10 pts] (GUI not necessary for this question)

Implement the **contains(val, col_name, table_name)** function inside server.js. This function checks whether the **table_name** table contains any tuple with a cell value equal to **val** in the **col_name** column. According to this, the function returns True or False.

Example evaluations:

- **contains("AFK", "countryCode", "city")** should return False.
- **contains("AFG", "countryCode", "city")** should return True.

Question 2. [20 pts]

(a) Design an Input-Output GUI (your GUI should be in q2.html) similar to the following (first one is for input; second one is for output):

First Country:

Second Country:

Languages
Arabic
Hindi

(b) Implement the **diff_lang(country1, country2)** function in the files server.js and middle.js. You should:

1. Read user inputs from the above GUI in middle.js and send a request to server.js for further processing.
2. Find the languages that are spoken in **country1** but not in **country2**. This should be done in server.js.
3. Print these languages to the above GUI as output. This should be done in middle.js.

You can use the examples from the PS to see how the link between server.js and middle.js should be established.

Important: Your **diff_lang** implementation MUST use a NESTED SQL QUERY. You are not allowed to write more than one query.

Example evaluations:

- **diff_lang**("Turkey", "United Arab Emirates"): In our database, the languages spoken in Turkey are "Arabic", "Kurdish", "Turkish". The languages spoken in the United Arab Emirates are "Arabic" and "Hindi". The result must be "Kurdish" and "Turkish".
- **diff_lang**("Turkey", "United Kingdom"): The result must be "Arabic", "Kurdish", "Turkish".

Question 3. [20 pts] Implement the **diff_lang_join(country1, country2)** function inside `server.js` and `middle.js`. To receive user's inputs and display the output, you should implement a GUI that is identical to Question 2.

diff_lang_join should return the same result as **diff_lang** in Question 2, but this time, you are **NOT** allowed to use nested SQL queries and/or keywords such as EXCEPT, IN, EXISTS, NOT IN, NOT EXISTS. Instead, you must use JOIN operations. Also, you are not allowed to write more than one query.

Question 4. [20 pts]

(a) Design an Input-Output GUI (your GUI should be in `q4.html`) similar to the following (first one is for input; second one is for output):

Choose an operation type: MIN ▾

Country Name:

Get Life Expectancy

Name	LifeExpectancy	GovernmentForm	Language
Afghanistan	45.9	Islamic Emirate	Dari
Afghanistan	45.9	Islamic Emirate	Pashto
Armenia	66.4	Republic	Armenian
Antigua and Barbuda	70.5	Constitutional Monarchy	English

(b) Implement the **aggregate_countries(agg_type, country_name)** function inside `server.js` and `middle.js`. You should:

1. Read user inputs from the above GUI in `middle.js` and send a request to `server.js` for further processing.
2. Find the countries that have higher life expectancy than **agg_type** life expectancy of all countries and lower life expectancy than **country_name**. This should be done in `server.js`.

- Print the Name, Life Expectancy, Government Type and Official Language of such countries to the GUI as output. This should be done in middle.js.

Similar to Question 2, you can use the examples from the PS to see how the link between server.js and middle.js should be established.

Among the aggregate operators, you only need to consider **agg_type** = **MIN** or **AVG**.

Example evaluations:

- aggregate_countries("AVG", "Turkey")** returns countries which have higher life expectancy than **average** life expectancy of all countries and lower life expectancy than **Turkey**.

Name	LifeExpectancy	GovernmentForm	Language
Antigua and Barbuda	70.5	Constitutional Monarchy	English
Bulgaria	70.9	Republic	Bulgarian
Belarus	68.0	Republic	Belorussian
Belarus	68.0	Republic	Russian
Belize	70.9	Constitutional Monarchy	English
Colombia	70.3	Republic	Spanish
Cape Verde	68.9	Republic	Portuguese
Algeria	69.7	Republic	Arabic
Estonia	69.5	Republic	Estonian
Fiji Islands	67.9	Republic	Fijian
Greenland	68.1	Part of Denmark	Danish
Greenland	68.1	Part of Denmark	Greenlandic
Honduras	69.9	Republic	Spanish
Indonesia	68.0	Republic	Malay
Iran	69.7	Islamic Republic	Persian
Iraq	66.5	Republic	Arabic
Saint Kitts and Nevis	70.7	Constitutional Monarchy	English
Lithuania	69.1	Republic	Lithuanian
Latvia	68.4	Republic	Latvian
Morocco	69.1	Constitutional Monarchy	Arabic
Mongolia	67.3	Republic	Mongolian
Malaysia	70.8	Constitutional Monarchy, Federation	Malay
Nicaragua	68.7	Republic	Spanish
Peru	70.0	Republic	Aimará
Peru	70.0	Republic	Ketšua
Peru	70.0	Republic	Spanish
Philippines	67.5	Republic	Pilipino
Palau	68.6	Republic	English
Palau	68.6	Republic	Palau
North Korea	70.7	Socialistic Republic	Korean
Romania	69.9	Republic	Romani
Romania	69.9	Republic	Romanian
Russian Federation	67.2	Federal Republic	Russian
Saudi Arabia	67.8	Monarchy	Arabic
El Salvador	69.7	Republic	Spanish
Seychelles	70.4	Republic	English
Seychelles	70.4	Republic	French
Syria	68.5	Republic	Arabic
Thailand	68.6	Constitutional Monarchy	Thai
Tonga	67.9	Monarchy	English
Tonga	67.9	Monarchy	Tongan
Vietnam	69.3	Socialistic Republic	Vietnamese
Samoa	69.2	Parlementary Monarchy	English
Samoa	69.2	Parlementary Monarchy	Samoaan

- aggregate_countries("MIN", "France")** returns countries which have higher life expectancy than **minimum** life expectancy of all countries and lower life expectancy than **France**.

Name	LifeExpectancy	GovernmentForm	Language
Aruba	78.4	Nonmetropolitan Territory of The Netherlands	Dutch
Afghanistan	45.9	Islamic Emirate	Dari
Afghanistan	45.9	Islamic Emirate	Pashto
Anguilla	76.1	Dependent Territory of the UK	English
Albania	71.6	Republic	Albaniana
Netherlands Antilles	74.7	Nonmetropolitan Territory of The Netherlands	Dutch
Netherlands Antilles	74.7	Nonmetropolitan Territory of The Netherlands	Papiamentu
United Arab Emirates	74.1	Emirate Federation	Arabic
Argentina	75.1	Federal Republic	Spanish
Armenia	66.4	Republic	Armenian
American Samoa	75.1	US Territory	English
American Samoa	75.1	US Territory	Samoan
Antigua and Barbuda	70.5	Constitutional Monarchy	English
Austria	77.7	Federal Republic	German
Azerbaijan	62.9	Federal Republic	Azerbaijani
Burundi	46.2	Republic	French
Burundi	46.2	Republic	Kirundi
Belgium	77.8	Constitutional Monarchy, Federation	Dutch
Belgium	77.8	Constitutional Monarchy, Federation	French
Belgium	77.8	Constitutional Monarchy, Federation	German
Bangladesh	60.2	Republic	Bengali
Bulgaria	70.9	Republic	Bulgariana
Bahrain	73.0	Monarchy (Emirate)	Arabic
Bosnia and Herzegovina	71.5	Federal Republic	Serbo-Croatian
Belarus	68.0	Republic	Belorussian
Belarus	68.0	Republic	Russian
Belize	70.9	Constitutional Monarchy	English
Bermuda	76.9	Dependent Territory of the UK	English
Bolivia	63.7	Republic	Aimará
Bolivia	63.7	Republic	Ketšua
Bolivia	63.7	Republic	Spanish
Brazil	62.9	Federal Republic	Portuguese
Barbados	73.0	Constitutional Monarchy	English
Brunei	73.6	Monarchy (Sultanate)	Malay
Bhutan	52.4	Monarchy	Dzongkha
Chile	75.7	Republic	Spanish
China	71.4	People's Republic	Chinese
Cook Islands	71.1	Nonmetropolitan Territory of New Zealand	Maori
Colombia	70.3	Republic	Spanish
Comoros	60.0	Republic	Comorian
Cape Verde	68.9	Republic	Portuguese
Costa Rica	75.8	Republic	Spanish
Cuba	76.2	Socialistic Republic	Spanish
Cyprus	76.7	Republic	Greek
Cyprus	76.7	Republic	Turkish
Czech Republic	74.5	Republic	Czech
Germany	77.4	Federal Republic	German
Djibouti	50.8	Republic	Arabic
Denmark	76.5	Constitutional Monarchy	Danish

Question 5. [30 pt] (GUI not necessary for this question)

(a) Implement the **find_min_max_continent()** function inside server.js. This function should find the name of the countries that have minimum and maximum life expectancy in each continent.

Example evaluation:

Name	Continent	LifeExpectancy
Afghanistan	Asia	45.9
Andorra	Europe	83.5
Australia	Oceania	79.8
Brazil	South America	62.9
Canada	North America	79.4
French Guiana	South America	76.1
Haiti	North America	49.2
Kiribati	Oceania	59.8
Macao	Asia	81.6
Moldova	Europe	64.5
Saint Helena	Africa	76.8
Zambia	Africa	37.2

(b) Implement the **find_country_languages(percentage, language)** function in server.js. This function should find the names of the countries that speak the **language** with more than **percentage** percent.

Example evaluations:

- **find_country_language(85, "Turkish")** returns countries in which more than 85% of the citizens speak Turkish.

Name	Language	Percentage
Turkey	Turkish	87.6

(c) Implement the **find_country_count(amount)** function inside server.js. This function should find the following: Let T denote the list of countries that have more than **amount** cities. For each continent, find which country among those in T has the maximum life expectancy in that continent. Print the name of the country that was found, the maximum life expectancy, and the continent.

Example evaluations:

- **find_country_count(100)** should return the following result:

Name	LifeExpectancy	Continent
Brazil	62.9	South America
Japan	80.7	Asia
Russian Federation	67.2	Europe
United States	77.1	North America

Submission

Your submission should consist of all of your JS and HTML files, zipped as: '**<your_student_ID>.zip**'.