**Project Describtion:**

**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):

(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):

(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.
3. 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**.

● **aggregate\_countries**(“**MIN**”, “**France**”) returns countries which have higher life expectancy than **minimum** life expectancy of all countries and lower life expectancy than **France**.

**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.

(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.

(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.

Fort he Project below i have class above. Can you update the classes?

Middle.js:

//fill the file

Server.js:

const express = require('express');

const bodyParser = require('body-parser');

const cors = require('cors');

const mysql = require('mysql');

const app = express();

// Enable CORS

app.use(cors());

// Configure body-parser to handle POST requests

app.use(bodyParser.urlencoded({ extended: true }));

app.use(bodyParser.json());

// Create a MySQL connection pool

const pool = mysql.createPool({

host: '...',

user: '...',

password: '...',

database: '...',

});

function contains(val, col\_name, table\_name){

}

export function diff\_lang(country1, country2){

}

export function diff\_lang\_join(country1, country2){

}

export function aggregate\_countries(agg\_type, country\_name) {

}

function find\_min\_max\_continent() {

}

function find\_country\_languages(percentage, language) {

}

function find\_country\_count(amount) {

}

app.get('/getDiffLang', (req, res) => {

diff\_lang(country1, country2 , (error, results) => {

});

});

app.get('/getDiffLangJoin', (req, res) => {

diff\_lang\_join(country1, country2 , (error, results) => {

});

});

app.get('/aggregateCountries', (req, res) => {

aggregate\_countries(agg\_type, country\_name , (error, results) => {

});

});

Q2.html:

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<link rel="stylesheet" href="Dashboard.css">

<title>COMP306</title>

<script type="text/javascript" src="middle.js"></script>

</head>

<body>

<h1>Welcome To Question 2</h1>

</body>

</html>

Q4.html:

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>COMP306</title>

<script type="text/javascript" src="middle.js"></script>

</head>

<body>

<h1>Welcome To Question 4</h1>

</body>

</html>