

Unit Test

This is the table from the database that I will use to do the unit tests from using XAMPP:

Table	Action	Rows	Type	Collation	Size	Overhead
<input type="checkbox"/> city	★ Browse Structure Search Insert Empty Drop	4,079	InnoDB	utf8mb4_general_ci	512.0 KiB	-
<input type="checkbox"/> country	★ Browse Structure Search Insert Empty Drop	239	InnoDB	utf8mb4_general_ci	112.0 KiB	-
<input type="checkbox"/> countrylanguage	★ Browse Structure Search Insert Empty Drop	984	InnoDB	utf8mb4_general_ci	160.0 KiB	-
3 tables	Sum	5,302	InnoDB	utf8mb4_general_ci	784.0 KiB	0 B

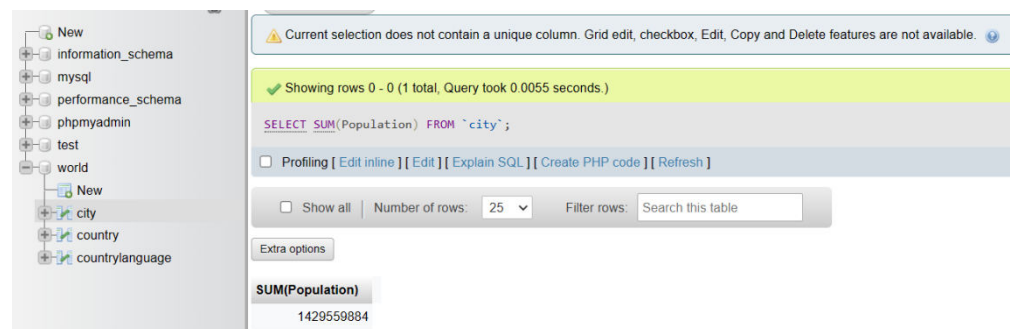
Unit Test 1

Name: Testing mathematical operations from the database using the sum function
From: World. SQL

Description: The first unit test I would do is to find the total number of population of people who lives in every city of each country in the database.

Screenshots of testing

```
SELECT SUM(Population) FROM `city`;
```



The screenshot shows a database management tool interface. On the left, there is a sidebar with a tree view of databases and tables. The main area displays the SQL query: `SELECT SUM(Population) FROM `city`;`. Below the query, there is a status bar indicating 'Showing rows 0 - 0 (1 total, Query took 0.0055 seconds.)'. The result of the query is shown as a single row with the value '1429559884' under the column 'SUM(Population)'. The interface also includes a 'Filter rows' section with a search bar and a 'Number of rows' dropdown set to 25.

Result: After I have completed the unit testing, I have found out the total population of people living in every city of each of the country from the database as the result was shown to be around 1.4 billion people.

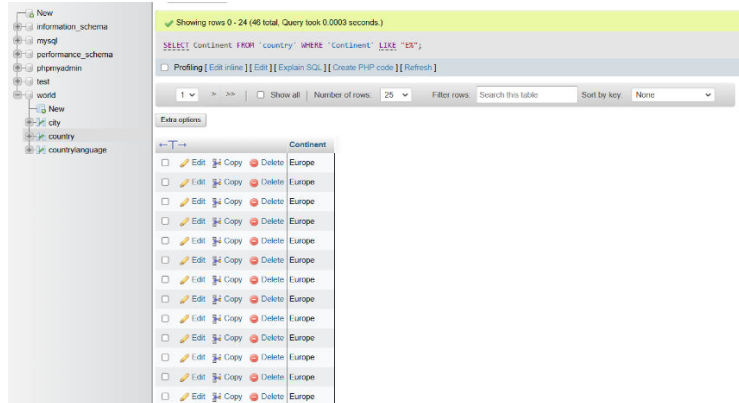
Unit Test 2

Name: Testing mathematical operations from the database using a special operator
From: World .SQL

Description: This unit test I would decide to do is to find the continent of the country in the world that is Europe that is only starting with the letter “E” to only filter this type of information in the database to show the right answer.

Screenshots of testing

```
SELECT * FROM `country`  
WHERE `Continent` LIKE "E%";
```



Result: After I have completed the unit testing, I have found out the continent of each country only contains Europe in the database as it begins with the letter “E”, so it does not include other continents of each country that are Asia, North America, South America, Africa, and Oceania.

Unit Test 3

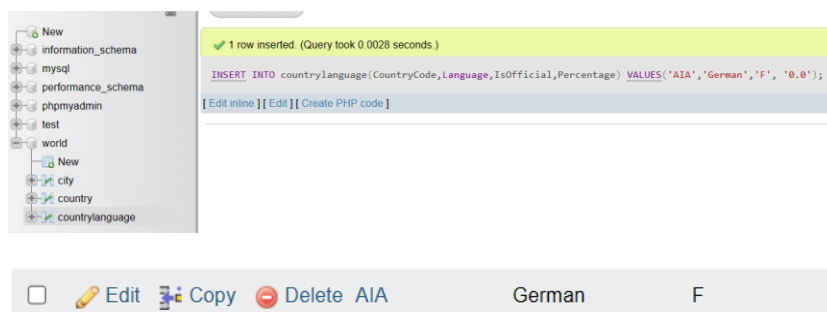
Name: To test database operations by inserting records in the database

From: World. SQL

Description: For this unit testing, I want to add a data to the database of the country Anguilla as the German language that is not official, and no one speaks the language from there.

Screenshot of testing

```
1 INSERT INTO countrylanguage(CountryCode,Language,IsOfficial,Percentage)  
2 VALUES('AIA','German','F','0.0');
```



Result: After I have completed this testing, I have found out that I have inserted a new record into the database that represents the country Anguilla as the country code, German as the language which is not official, and it is a British territory, so no percentage of the people speak German only English.

Unit Test 4

Name: To test database operations by updating records in the database
From: World. SQL

Description: For this unit testing, I am going to update the population of people who are living in a city in the country of Netherlands called Utrecht in the database.

Screenshot of testing

```
1 UPDATE `city`
2 SET Population = 565000
3 WHERE ID = 8;
```

1 row affected. (Query took 0.0044 seconds.)

```
UPDATE `city` SET Population = 565000 WHERE ID = 8;
```

[Edit inline] [Edit] [Create PHP code]

<input type="checkbox"/>	Edit	Copy	Delete	8 Utrecht	NLD	Utrecht	234323
<input type="checkbox"/>	Edit	Copy	Delete	8 Utrecht	NLD	Utrecht	565000

Result: After I have completed this testing, I have found out that I have updated the population of the people who were living in Utrecht. The population increased in that city from a few years ago that contained the data that was recorded before and now it is the present data that represents the population for now.

Unit Test 5

Name: To test the error handling from the input validation in the database
From: World. SQL

Description: I am going to create a new table about the United Kingdom by adding the values as attributes for it in the database.

Screenshot of testing

```
1 INSERT INTO city(ID,Name,CountryCode,District,Population)
2 VALUES('5000','United Kingdom','GBR', 'England', 'sixty-seven million');
```

1 row inserted. (Query took 0.0088 seconds.)

```
INSERT INTO city(ID,Name,CountryCode,District,Population) VALUES('5000','United Kingdom','GBR', 'England', 'sixty-seven million');
```

[Edit inline] [Edit] [Create PHP code]

Warning: #1366 Incorrect integer value: 'sixty-seven million' for column 'world`.`city`.`Population' at row 1

	ID	Name	CountryCode	District	Population
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	5000	United Kingdom	GBR	England	0

Result: After the testing, I have managed to create a new table for the city name as the UK which should have been as a country and I typed to insert the values of it as attributes, but the warning message was shown after I created the table which was an error as for the population I have entered the value as a string instead of an integer, so the result of that was the population contained no value.

Unit Test 6

Name: To test the database operation by deleting a record in the database

From: World. SQL

Description: For this unit test, I will delete a record that I have made from the previous test as it contained an error before so that it does not appear in the database.

Screenshot of testing

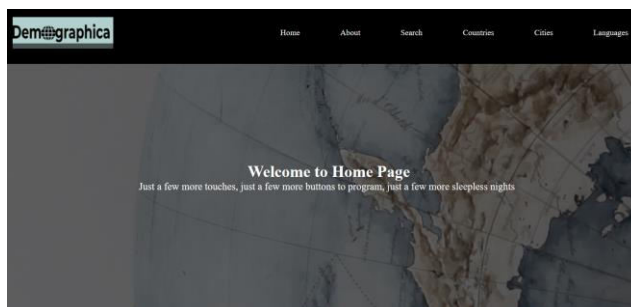
	ID	Name	CountryCode	District	Population
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	5000	United Kingdom	GBR	England	0

Confirm
☐

Do you really want to execute "DELETE FROM city WHERE `city`.`ID` = 5000"?

Result: After the testing, I have selected the specific record of data to be removed from the database which was deleted after I have confirmed it. This means that this deleted record from the database will not appear anymore unless I have created the table again.

This is the website I will use to do some more unit testing:



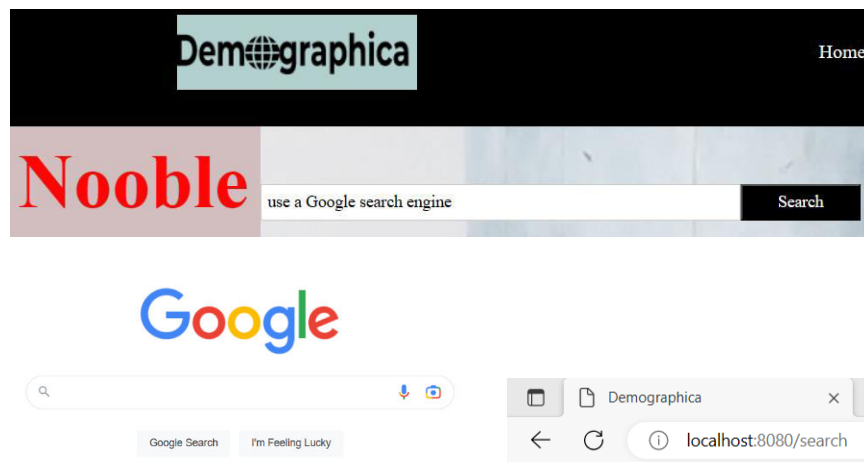
Unit Test 7

Name: To test basic functions on the website

From: docker-compose.yml

Description: For this unit test, I will test the search engine in the search section of the website to see if it is working when I look to type anything inside the search engine to browse.

Screenshot of testing



Result: After the testing, I have entered some words in the website's search engine optimisation that is called Nooble and pressed the search button to start. After I pressed the search button, the website automatically took me to the Google's search engine rather than the result to be shown as an outcome on the website, but the search button works on the website, and it uses the Google search engine to type words and provide answers online as the website is localhost.

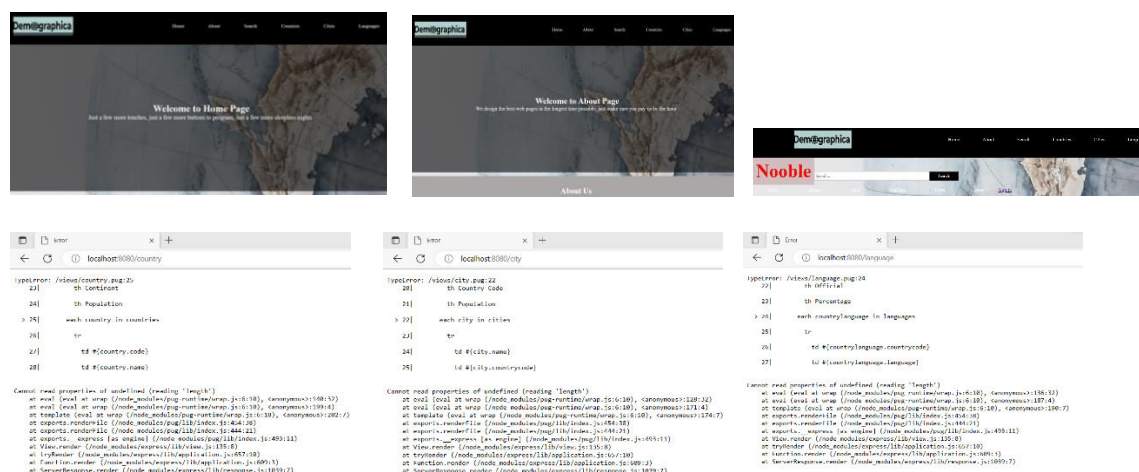
Unit Test 8

Name: To test more basic functions on the website

From: docker-compose.yml

Description: For this unit test, I am going to test to see if all the buttons for the features of the website is working which are the home, about, search, countries, cities, and languages pages when I click on every part of the website to load with its contents to be seen.

Screenshot of testing



Result: After the testing, I have found out that some of the web pages when I clicked on them to see its content to know the features on the website. The web pages worked for the home, about and search pages. But the errors were shown when I clicked on the countries, cities, and languages pages from my laptop that I was using to run the website.

