令和 7 年度

実 験 レ ポ ー ト

題目	データベースの構築

学　　科	電気電子システム工学コース
学籍番号	a0527
氏　　名	野口 史遠
提 出 日	令和 7 年 6 月 9 日

National Institute of Technology, Maizuru College
NIT, Maizuru KYOTO

舞鶴工業高等専門学校

目 次

実験目的

本実験の目的は，様々なシステムの根幹をなすデータベースの基礎的な概念と構築方法を学習し，Web データベースの作成および運用ができるようになることである．具体的には，XAMPP を用いたローカルサーバ環境を構築し，MySQL（MariaDB）によるデータベースの作成，PHP によるデータ操作，および Google Maps API を活用した Web インターフェースの構築を通じて，実践的な Web データベースの仕組みを理解する．

第 1 章　理論

本実験では，Web 技術とデータベース技術を連携させた Web データベースを構築する．これにより，ユーザは Web ブラウザを通じて地理情報を直感的に扱うことが可能となる．以下に，各技術要素の概要を示す．

1.1　データベース

データベースとは，情報を一定の形式で蓄積・管理し，効率的に検索・更新できるようにする仕組みである．データを一元的に保存することにより，情報の重複や不整合を防ぎ，複数のユーザ間での共有や同時利用が可能となる．

データベースの管理には，データベース管理システム（DBMS: Database Management System）が用いられる．DBMS は，データの記録・更新・検索・削除といった操作を統一的なインターフェースで提供するソフトウェアであり，ユーザやアプリケーションプログラムは DBMS を介してデータベースにアクセスする．

特に広く用いられているのがリレーショナルデータベース管理システム（RDBMS: Relational DBMS）である．これは，データをテーブル形式で構造化し，各行がレコード，各列がフィールドとして定義される．RDBMS では SQL（Structured Query Language）と呼ばれる言語を用いて，データの追加・取得・更新・削除といった操作を行う．

本実験では，MySQL と互換性のある MariaDB を使用し，「施設名」「住所」「緯度・経度」「施設の種別」などの地理情報を扱うテーブルを作成し，Web アプリケーションから動的に操作可能な環境を構築する．

1.2　Web データベース

Web データベースとは，インターネットを介して Web ブラウザからアクセス・操作が可能なデータベースシステムであり，ユーザが PC やスマートフォンなどの端末を用いて，データの検索や登録，更新などの操作を視覚的かつ直感的に行える環境を提供する．Web データベースは，通常，Web サーバ上で動作するアプリケーションとデータベース管理システム（DBMS）を組み合わせて構築される．本実験では，PHP で記述したスクリプトを介して，Web ページ上のユーザインターフェースと MariaDB を連携させることにより，動的に扱える Web アプリケーションの構築を行う．

1.3　Google Maps API

Google Maps API は，Web サイト上で地図情報を表示，操作できる JavaScript ベースの API である．緯度・経度の情報をもとにマーカーを地図上に表示したり，マーカーをクリックして情報ウィンドウを開くなどのインタラクションが可能となる．

本実験では，Google Maps JavaScript API を用いて，データベースに保存された地点情報（施設名，住所，緯度・経度，種別など）を読み込み，マーカーとして地図上に表示する．各マーカーには施設の名称や詳細情報が紐づけられており，クリックすることで情報ウィンドウが表示される．

第 2 章　実験方法

本実験では，段階的に Web データベースを構築するため，複数のステップに分けて作業を行った．ここでは，まず最初に行った XAMPP のインストールおよび動作確認について記述する．

環境

- OS：Ubuntu 22.04 LTS

実験 I：XAMPP のインストールと起動確認

XAMPP は，Apache Web サーバ，MySQL（MariaDB），PHP，Perl などを一括でインストール・管理できるパッケージである．本実験では，XAMPP をインストールして Apache の起動確認を行った．

手順は以下の通りである．

1. 以下の公式サイトより，XAMPP の Linux 版インストーラをダウンロードする．`https://www.apachefriends.org/jp/index.html`
2. ターミナルでダウンロードディレクトリに移動する．

```
cd ~/Downloads/
```

3. インストーラに実行権限を与える．

```
sudo chmod 755 xampp-linux-x64-8.0.10-0-installer.run
```

4. インストーラを実行する．

```
sudo ./xampp-linux-x64-8.0.10-0-installer.run
```

5. GUI ベースのマネージャを起動する．

```
sudo /opt/lampp/manager-linux-x64.run
```

6. LAMP 環境を起動する．

```
sudo /opt/lampp/lampp start
```

7. ブラウザで `http://localhost/` を開き，「XAMPP へようこそ」と表示されれば Apache の動作確認は完了である．
8. 確認後，LAMP 環境を停止する．

```
sudo /opt/lampp/lampp stop
```

この操作により，Web サーバの動作環境が整い，今後の Web データベース構築に必要な基盤が準備された．

図 2.1 : Welcome to XAMPP

実験 II：Web ページ作成とデータベース構築

本実験では, HTML による Web ページの作成と, phpMyAdmin を用いたデータベース（MariaDB）の作成・検索，および PHP を用いた Web データベースの構築を行った.

1. Web ページの作成

XAMPP のドキュメントルートに移動し，HTML ファイルを作成した.

```
cd /opt/lampp/htdocs/
```

作成したファイル `sample.html` は以下の通りである.

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<title>こんにちは</title>
</head>
<body>
<h1>見出し</h1>
<p>文章を書きましょう</p>
</body>
</html>
```

これをブラウザで `http://localhost/sample.html` にアクセスすることで表示を確認した.

次に，CSS によって文字色を赤く変更したスタイルを追加した.

```
<style>
 p {
   color: red;
 }
</style>
```

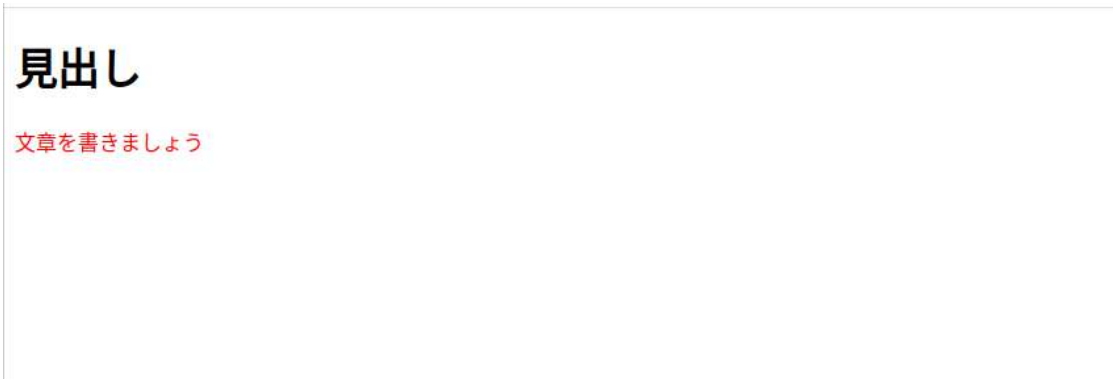

図 2.2 ：sample.html

2. データベースの作成・検索

`phpMyAdmin` を用いて，データベース `maizuru` を作成し，以下の 2 つのテーブルを構築した.

- **name テーブル**：no, name（TEXT 型）
- **birthplace テーブル**：name, birthplace（TEXT 型）

データを追加後，以下の SQL 文でデータを検索・表示した．

```
1  SELECT * FROM name;
```

図 2.3 ：maizuru の name テーブル

図 2.4 ：maizuru の birthplace テーブル

さらに，2 つのテーブルを結合して，対応する生誕地を表示するクエリを実行した．

```
1  SELECT name.no, name.name, birthplace.birthplace
2  FROM name
3  INNER JOIN birthplace
4  ON name.name = birthplace.name;
```

3. Web データベースの構築（PHP）

MariaDB のパスワードを設定するため，--skip-grant-tables モードで起動し，次のコマンドでパスワードを直接設定した．

```
sudo /opt/lampp/sbin/mysqld --skip-grant-tables --skip-networking &
sudo /opt/lampp/bin/mysql -u root

UPDATE mysql.user
SET authentication_string = PASSWORD('password')
WHERE User = 'root' AND Host = 'localhost';
FLUSH PRIVILEGES;
sudo killall mysqld
sudo /opt/lampp/lampp start
```

その後，PHP を用いた接続確認とデータ表示のため，以下のファイルを配置した．

connect.php

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<title>PHP TEST</title>
</head>
<body>

<?php

$link = mysqli_connect('localhost', 'root', 'password');
if (!$link) {
    die('接続失敗です。'.mysql_error());
}

print('<p接続に成功しました。></p>');

$db_selected = mysqli_select_db($link,'maizuru');
if (!$db_selected){
    die('データベース選択失敗です。'.mysql_error());
}

print('<pデータベースを選択しました。></p>');

// に対する処理 MySQL

$close_flag = mysqli_close($link);

if ($close_flag){
    print('<p切断に成功しました。></p>');
}

?>
</body>
</html>
```

select.php

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
```

接続に成功しました。

データベースを選択しました。

切断に成功しました。

図 2.5 ： connect.php

```
4   <title>PHP TEST</title>
5   </head>
6   <body>
7
8   <?php
9
10  $link = mysqli_connect('localhost', 'root', 'password');
11  if (!$link) {
12      die('接続失敗です。'.mysql_error());
13  }
14
15  print('<p接続に成功しました。></p>');
16
17  $db_selected = mysqli_select_db($link,'maizuru');
18  if (!$db_selected){
19      die('データベース選択失敗です。'.mysqli_error());
20  }
21
22  print('<pデータベースを選択しました。></p>');
23
24  mysqli_set_charset($link,'utf8');
25
26  $result = mysqli_query($link,'SELECT * FROM name');
27  if (!$result) {
28      die('クエリーが失敗しました。'.mysql_error());
29  }
30
31  while ($row = mysqli_fetch_assoc($result)) {
32      print('<p>');
33      print('id='.$row['no']);
34      print(', name='.$row['name']);
35      print('</p>');
36  }
37
38  $close_flag = mysqli_close($link);
39
40  if ($close_flag){
41      print('<p切断に成功しました。></p>');
```

```
42  }
43
44  ?>
45  </body>
46  </html>
```

接続に成功しました。

データベースを選択しました。

id=a0401, name=夏目漱石

id=a0402, name=太宰治

id=a0403, name=井上靖

id=a0404, name=井上ひさし

切断に成功しました。

図 2.6 : select.php

select2.php

```
1   <html>
2   <head>
3   <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
4   <title>PHP TEST</title>
5   </head>
6   <body>
7
8   <?php
9
10  $link = mysqli_connect('localhost', 'root', 'password');
11  if (!$link) {
12      die('接続失敗です。'.mysql_error());
13  }
14
15  print('<p接続に成功しました。></p>');
16
17  $db_selected = mysqli_select_db($link,'maizuru');
18  if (!$db_selected){
19      die('データベース選択失敗です。'.mysql_error());
20  }
21
22  print('<pデータベースを選択しました。></p>');
23
24  mysqli_set_charset($link,'utf8');
25
26  $result = mysqli_query($link,'SELECT name.no, name.name, birthplace.birthplace FROM name inner join
        birthplace on name.name=birthplace.name');
27  if (!$result) {
28      die('クエリーが失敗しました。'.mysqli_error());
29  }
```

```
30
31    while ($row = mysqli_fetch_assoc($result)) {
32        print('<p>');
33        print('id='.$row['no']);
34        print(', name='.$row['name']);
35        print(', birthplace='.$row['birthplace']);
36        print('</p>');
37    }
38
39    $close_flag = mysqli_close($link);
40
41    if ($close_flag){
42        print('<p切断に成功しました。></p>');
43    }
44
45    ?>
46    </body>
47    </html>
```

接続に成功しました。

データベースを選択しました。

id=a0401, name=夏目漱石, birthplace=東京都, year_of_birth=1867

id=a0402, name=太宰治, birthplace=青森県, year_of_birth=1909

id=a0403, name=井上靖, birthplace=北海道, year_of_birth=1907

id=a0404, name=井上ひさし, birthplace=山形県, year_of_birth=1934

切断に成功しました。

図 2.7 ： select2.php

form_name.html（入力フォーム）

```
1   <html>
2
3   <head>
4     <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
5     <title>検索</title>
6   </head>
7
8   <body>
9     <form action="regist.php" method="post">番号：
10      <br />
11      <input type="text" name="no" size="30" value="" /><br />名前：
12      <br />
13      <input type="text" name="name" size="30" value="" /><br />出身地：
14      <br />
15      <input type="text" name="birthplace" size="30" value="" /><br />生年：
16      <br />
17      <input type="text" name="year_of_birth" size="30" value="" /><br />
```

```
18      <br />
19      <input type="submit" value="登録する" />
20    </form>
21  </body>
22
23  </html>
```

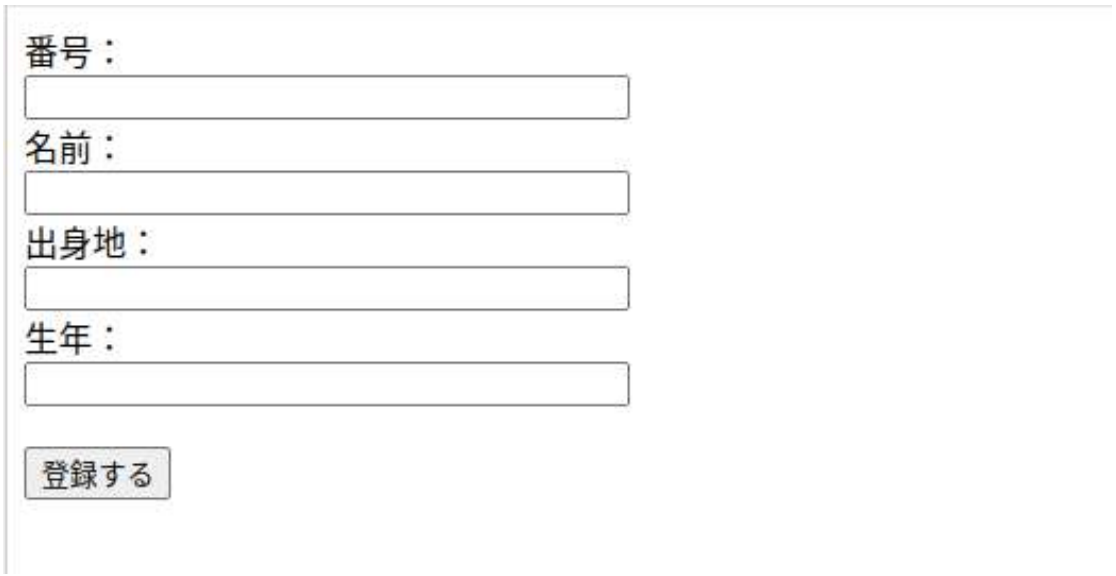

図 2.8 ： form_name.html

これにより，Web フォームからデータベースへ動的に登録し，登録結果を確認する一連の Web データベース操作が可能となった．

実験III：Google Maps API を活用した Web データベースの構築

本実験では，Google Maps API を活用して，データベースに登録された地理情報を地図上に可視化する Web アプリケーションを構築した．

1. データベースの作成

phpMyAdmin にて，データベース `map` を新規作成し，以下の構成で `markers` テーブルを作成した．

```
1  CREATE TABLE ‘markers‘ (
2    ‘id‘ INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
3    ‘name‘ VARCHAR(60) NOT NULL,
4    ‘address‘ VARCHAR(80) NOT NULL,
5    ‘lat‘ FLOAT(10,6) NOT NULL,
6    ‘lng‘ FLOAT(10,6) NOT NULL,
7    ‘type‘ VARCHAR(30) NOT NULL
8  ) ENGINE = MYISAM;
```

初期データとして以下のようなレコードを登録した．

```
INSERT INTO `markers` (`name`, `address`, `lat`, `lng`, `type`)
VALUES ('Love.Fish', '580 Darling Street, Rozelle, NSW', -33.861034, 151.171936, 'restaurant');
```

2. データ出力用 PHP スクリプト

登録されたデータを XML 形式で出力するため，`toxml.php` を作成した．このスクリプトは，SQL クエリにより `markers` テーブルから全レコードを取得し，Google Maps JavaScript API が読み取れる形式の XML に変換して出力する．

3. Google Maps API による表示ページの構築

`gis01.html`，`gis02.html`，`index.html` を作成し，Google Maps 上にマーカーとしてデータを描画した．

gis01.html　基本的な地図表示を行うページであり，以下のように初期表示位置とズームレベルを設定している：

```
map = new google.maps.Map(document.getElementById('map'), {
  center: {lat: -34.397, lng: 150.644},
  zoom: 8
});
```

gis02.html　`toxml.php` から取得した XML を解析し，地図上に各マーカーを配置する．

index.html　ユーザが地図上で検索できる UI を提供する．JavaScript（`script.js`）により，検索ボックスやマーカーの追加が行われ，より直感的な操作が可能となる．

4. JavaScript によるマーカー描画（script.js）

Google Maps API の `places` ライブラリと非同期通信（AJAX）を利用し，XML データを解析して地図上にマーカーを配置する処理は以下の通りである．

```
downloadUrl('toxml.php', function(data) {
  var xml = data.responseXML;
  var markers = xml.documentElement.getElementsByTagName('marker');
  ...
  var marker = new google.maps.Marker({
    map: map,
    position: point,
    label: icon.label
  });
```

マーカーをクリックすると，施設名と住所を表示する情報ウィンドウが開くように設定している．

第 3 章　課題

課題 1：XAMPP のインストールと起動確認

本課題では，XAMPP のインストールおよび Apache の起動確認を行う．この作業は実験 I にて完了済みであるため，ここでは割愛する．

課題 2：新しいデータベースの作成と表示の修正

1. year_of_birth テーブルの追加

既存の name および birthplace テーブルに加えて，新たに year_of_birth テーブルを作成し，生年情報を管理する．

図 3.1 ：maizuru の year_of_birth テーブル

2. PHP コードの修正

3 つのテーブル name，birthplace，year_of_birth を結合し，生年情報も含めて表示するように select.php を修正した．

```
1  <html>
2  <head>
3  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
4  <title>PHP TEST</title>
5  </head>
6  <body>
7  <?php
8  $link = mysqli_connect('localhost', 'root', 'password');
9  if (!$link) {
```

```
    die('接続失敗です。'.mysql_error());
}
print('<p接続に成功しました。></p>');

$db_selected = mysqli_select_db($link,'maizuru');
if (!$db_selected){
    die('データベース選択失敗です。'.mysql_error());
}
print('<pデータベースを選択しました。></p>');

mysqli_set_charset($link,'utf8');

$result = mysqli_query($link,'SELECT name.no, name.name, birthplace.birthplace, year_of_birth.year_of_birth
     FROM name INNER JOIN birthplace ON name.name = birthplace.name INNER JOIN year_of_birth ON name.name =
     year_of_birth.name');
if (!$result) {
    die('クエリーが失敗しました。'.mysqli_error());
}

while ($row = mysqli_fetch_assoc($result)) {
    print('<p>');
    print('id='.$row['no']);
    print(', name='.$row['name']);
    print(', birthplace='.$row['birthplace']);
    print(', year_of_birth='.$row['year_of_birth']);
    print('</p>');
}

$close_flag = mysqli_close($link);
if ($close_flag){
    print('<p切断に成功しました。></p>');
}
?>
</body>
</html>
```

これにより，新たに生年情報を含めた Web データベースの表示が可能となった．

課題 3：Google Maps API を用いた Web データベースの作成

1. markers テーブルへの登録

新たに 6 件の舞鶴市内の飲食店情報を登録した．

```
INSERT INTO `markers` (`id`, `name`, `address`, `lat`, `lng`, `type`) VALUES
('1', 'Dairokumaru', '580 Darling Street, Rozelle, NSW', '35.4490856', '135.3105046', 'seafood'),
('2', 'Maikei Kamiagu', '76 Wilford Street, Newtown, NSW', '35.4508761', '135.3312693', 'seafood'),
('3', 'Kirameki no Tori', '36 Blue St, North Sydney NSW', '35.4806006', '135.4229714', 'ramen'),
('4', 'NetsuretsuRamen', '2 Huntley Street, Alexandria, NSW', '35.4809550', '135.4257069', 'ramen'),
('5', 'Manma frypan', '43 Macpherson Street, Bronte, NSW', '35.4671916', '135.3943122', 'cafe'),
('6', 'cafe de BONO', '60-64 Reservoir Street, Surry Hills, NSW', '35.4761566', '135.3901868', 'cafe');
```

2. 地図初期位置の変更

舞鶴市中心の座標に地図の初期表示位置を設定するよう，JavaScript 関数 `initAutocomplete()` を修正した．

```
function initAutocomplete() {
  // 地図の初期設定
  map = new google.maps.Map(document.getElementById("map"), {
    center: { lat: 35.4761566, lng: 135.3901868 }, // 舞鶴の中心
    zoom: 13, // 初期ズームレベル
    mapTypeId: "roadmap",
  });
}
```

これにより，舞鶴市の飲食店情報を視覚的に確認できる Web 地図アプリケーションが実現した．

参 考 文 献

1) Google Cloud：Google Maps Platform Documentation, https://developers.google.com/maps/documentation（参照日：2025 年 6 月）.
2) Welling, L. and Thomson, L.：*PHP and MySQL Web Development*, Addison-Wesley Professional, 5th Edition, 2016.
3) 荻原 剛志：MySQL 徹底入門 第 3 版, 翔泳社, 2018 年.