

1. websystem基礎実験 MySQL編

今回使用するリポジトリは以下のとおりです。

<https://github.com/sudahiroshi/websystem2>

1.1. Paiza Cloudの起動

新規サーバ作成時にNode.jsとMySQLをクリック（タップ）で選択しておいてください。

1.2. MySQLの初期設定

1.2.1. mysqlコマンド

ユーザを作成するために、まずは管理者権限でログインする。コマンド名はそのものズバリmysqlである。

```
~$ sudo mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 4
Server version: 5.7.24-0ubuntu0.18.04.1 (Ubuntu)

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

1.2.2. ユーザ作成とパスワード設定

ユーザnodeを作成する。ユーザ名は各自変更して構わないが、後々のサンプルプログラムとの整合性を考えて、ひとまずnodeのまま作成すること。

```
mysql> create user 'node'@'localhost' identified with
mysql_native_password by 'pw';
Query OK, 0 rows affected (0.00 sec)

mysql>
```

続いて、nodeのパスワードを設定する。ここではwebsystemというパスワードにする。

```
mysql> set password for 'node'@'localhost' = password('websystem');
Query OK, 0 rows affected, 1 warning (0.01 sec)

mysql>
```

1.2.3. データベース作成と権限の付与

データベース`web`を作成する。例によってデータベース名は各自変更して構わないが、後々のサンプルプログラムとの整合性を考えて、ひとまず`web`のまま作成すること。

```
mysql> create database web;
Query OK, 1 row affected (0.00 sec)

mysql>
```

続いて、データベース`web`にアクセスする権限を、先ほど作成したユーザ`node`に付与する。

```
mysql> grant all on web.* to node@localhost;
Query OK, 0 rows affected (0.00 sec)

mysql>
```

1.2.4. ユーザやデータベースの確認

一応、きちんと作成できているか確認しよう。（と言っても、ユーザがきちんと作成されていなければ途中でエラーが発生する）

ユーザがきちんと作成されていれば、以下のように表示される。

```
mysql> show grants for 'node'@'localhost';
+-----+
| Grants for node@localhost |
+-----+
| GRANT USAGE ON *.* TO 'node'@'localhost' |
+-----+
1 row in set (0.00 sec)

mysql>
```

データベースを一覧した例を示す。上から4つはシステムが使用するデータベースであり、その下に先ほど作成した`web`が存在している。

```
mysql> show databases;
+-----+
```

```
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
| web |
+-----+
5 rows in set (0.01 sec)

mysql>
```

ここまでできていれば、初期設定は完了である。mysqlコマンドを終わらせるには、`exit`コマンドを使用する。

```
mysql> exit
Bye
~$
```

1.2.5. ここまでの設定を簡単に済ますために

毎回上記の設定を行うのはタイヘンなので、ここまでの作業をバックアップしたファイルを用意した。次回からは、そのファイルをリストアすれば良い。やり方は以下のとおりです。

```
~$ git clone https://github.com/sudahiroshi/websystem2.git
Cloning into 'websystem2'...
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (6/6), done.
~$ cd websystem2
~/websystem2$ sudo mysql < init.sql
~/websystem2$
```

1.3. テーブルの作成

mySQLはデータベースの中でも、リレーショナルデータベースと呼ばれる種類に属する。これは、複数のテーブルから構成されるデータを連結（リレーション）して使用するタイプのデータベースである。よって、使用する際には、テーブルの作成が必要となる。まずは、簡単なテーブルを作成してみよう。

まず、先ほど作成したユーザ`node`でmysqlにログインする。

```
~/websystem2$ mysql -u node -pwebsystem web
mysql: [Warning] Using a password on the command line interface can be insecure.
```

```
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 6
Server version: 5.7.24-0ubuntu0.18.04.1 (Ubuntu)

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights
reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input
statement.

mysql>
```

続いて、`create`コマンドを使用してテーブルを作成する。ここでは、名前が入力できるシンプルなテーブルを作成する。ここでは、テーブル名がnamesで、nameというvarchar（可変文字数の文字列）のみを持つテーブルを作る例である。

```
mysql> create table names ( name varchar(100) );
Query OK, 0 rows affected (0.07 sec)

mysql>
```

テーブルが作られていることを確認するには、`describe`コマンドを使用する。

```
mysql> describe names;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| name  | varchar(100) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql>
```

1.4. データの挿入

続いて、`insert`コマンドを使用してテーブルにデータを挿入する。ここでは、テーブル名namesの、列nameに、値ルフィを挿入している。

```
mysql> insert into names (name) values ('ルフィ');
Query OK, 1 row affected (0.01 sec)

mysql>
```

1.5. データの取り出し

続いて、`select`コマンドを使用してデータを取り出してみる。`*`は、すべての列から取り出すことを意味し、`from`の後ろでテーブル名を指定している。

```
mysql> select * from names;
+-----+
| name   |
+-----+
| ルフィ |
+-----+
1 row in set (0.00 sec)

mysql>
```

1.6. テーブルにデータを増やす

ここまで理解できたら、`insert`コマンドを用いてもう数件のデータを加えて、`select`コマンドで内容を確認しよう。好きな名前を追加して構わない。追加した例を以下に示す。

```
mysql> select * from names;
+-----+
| name   |
+-----+
| ルフィ |
| ゾロ   |
| ナミ   |
+-----+
3 rows in set (0.00 sec)

mysql>
```

1.7. データの上書き

リレーショナルデータベースを使う上でほとんどの処理は`insert`と`select`が用いられる。その他にはデータの上書きをする`update`とデータの削除を行う`delete`があるので、順に使ってみる。

まずは`update`を使用して上書きをしてみよう。書式は、`update テーブル名 set 列名=値 where 列名=値;`である。ここで、`where`を指定しなかった場合は、すべてのデータに同じ内容が上書きされるので注意すること。

```
mysql> update names set name='モンキー・D・ルフィ' where name='ルフィ';
Query OK, 1 row affected (0.02 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from names;
+-----+
| name                |
+-----+
| モンキー・D・ルフィ |
| ゾロ                |
| ナミ                |
+-----+
3 rows in set (0.00 sec)

mysql>
```

1.8. データの削除

続いて`delete`コマンドを使用してデータを削除してみよう。書式は`delete from テーブル名 where 列名=値;`である。当然、`where`を指定しなかった場合はすべてのデータが削除されるので注意すること。

```
mysql> delete from names where name='ナミ';
Query OK, 1 row affected (0.01 sec)

mysql> select * from names;
+-----+
| name                |
+-----+
| モンキー・D・ルフィ |
| ゾロ                |
+-----+
2 rows in set (0.00 sec)

mysql>
```

1.9. データを集める

基本的な操作を理解したと思うので、実際にデータを入れてデータベースとして使用してみる。本来は複数のテーブルから構成されるデータ群を連結（リレーション）するのだが、まずは1枚のテーブルのみを扱う。

なお、ここで使用するデータは政府の統計データを掲載しているサイト [e-Stat](#) からダウンロードした。今回は演習授業用に、都道府県別の人口統計と大学のデータを別々にダウンロードして結合したものを使用する。ただし、年度がずれているのでその点だけご了承ください。

1.9.1. テーブルを作る

データを読み込むにあたって、まずはテーブルexampleを作成する。項目は以下の表のとおりとする。

列名	型	説明
id	int	通し番号
都道府県	varchar(100)	都道府県名
人口	int	人口
男性	int	男性の人口
女性	int	女性の人口
大学	int	大学数
国立大学	int	国立大学数
公立大学	int	公立大学数
私立大学	int	私立大学数
学生数	int	学生数
男子学生	int	男子学生数
女子学生	int	女子学生数

表に従って、createコマンドを実行する。ちょっと長いので注意すること。なお、コマンドの途中で改行しているが、実際には改行してもしなくても良い。

```
mysql> create table example (  
  id int auto_increment not null primary key,  
  都道府県 varchar(100),  
  人口 int,  
  男性 int,  
  女性 int,  
  大学 int,  
  国立大学 int,  
  公立大学int,  
  私立大学 int,  
  学生数 int,  
  男子学生 int,  
  女子学生 int );  
Query OK, 0 rows affected (0.03 sec)  
  
mysql> describe example;  
+-----+-----+-----+-----+-----+-----+  
| Field      | Type          | Null | Key | Default | Extra          |  
+-----+-----+-----+-----+-----+-----+  
| id         | int(11)       | NO   | PRI | NULL    | auto_increment |  
| 都道府県   | varchar(100)  | YES  |     | NULL    |                |  
| 人口       | int(11)       | YES  |     | NULL    |                |  
| 男性       | int(11)       | YES  |     | NULL    |                |  
| 女性       | int(11)       | YES  |     | NULL    |                |
```

```

| 大学          | int(11)      | YES |      | NULL |
| 国立大学      | int(11)      | YES |      | NULL |
| 公立大学      | int(11)      | YES |      | NULL |
| 私立大学      | int(11)      | YES |      | NULL |
| 学生数        | int(11)      | YES |      | NULL |
| 男子学生      | int(11)      | YES |      | NULL |
| 女子学生      | int(11)      | YES |      | NULL |
+-----+-----+-----+-----+-----+
12 rows in set (0.02 sec)

mysql>

```

1.9.2. ファイルからデータを読み込む

都道府県別のデータを手入力するのは難しいので、こちらで用意したファイル`example.csv`を`load`コマンドを用いて読み込む。なお、`enclosed by`の後ろは「シングルクォーテーション」「ダブルクォーテーション」「シングルクォーテーション」の3文字なので注意すること。

```

mysql> load data local infile 'example.csv' into table example
fields terminated by ',' enclosed by '"'
(都道府県, 人口, 男性, 女性, 大学, 国立大学, 公立大学, 私立大学, 学生数, 男子学生, 女子学生 );
Query OK, 47 rows affected (0.04 sec)
Records: 47  Deleted: 0  Skipped: 0  Warnings: 0

mysql>

```

1.9.3. 項目を限定したデータの取得

続いて、データを確認するが、データ数が多くなっていて見づらいので、以下のように取得データ数に制限をかけて確認しよう。`select`の後ろの、先程まで*を書いていた箇所は、列名を書ける。複数の列を指定したい場合は**半角のカンマ**で区切ってつなげることが可能である。また、データ数は`limit`の後ろの数字で調整できる。

```

mysql> select id, 都道府県, 人口 from example limit 10;
+----+-----+-----+
| id | 都道府県 | 人口 |
+----+-----+-----+
| 1  | 北海道   | 5286000 |
| 2  | 青森県   | 1263000 |
| 3  | 岩手県   | 1241000 |
| 4  | 宮城県   | 2316000 |
| 5  | 秋田県   | 981000  |
| 6  | 山形県   | 1090000 |
| 7  | 福島県   | 1864000 |
| 8  | 茨城県   | 2877000 |
| 9  | 栃木県   | 1946000 |
| 10 | 群馬県   | 1952000 |

```



```
+-----+-----+-----+
10 rows in set (0.00 sec)

mysql>
```

1.9.4. 並べ替え

この`select`コマンドは多くの機能を持っており、Webアプリケーションを作成する際に使いこなすことが重要となってくる。続いて、並べ替えて表示してみよう。まずは人口の多い順と少ない順に10項目ずつ表示する例である。`order by`という記述に注目すること。

```
mysql> select id, 都道府県, 人口 from example order by 人口 desc limit 10;
+-----+-----+-----+
| id | 都道府県 | 人口 |
+-----+-----+-----+
| 13 | 東京都 | 13822000 |
| 14 | 神奈川県 | 9177000 |
| 27 | 大阪府 | 8813000 |
| 23 | 愛知県 | 7537000 |
| 11 | 埼玉県 | 7330000 |
| 12 | 千葉県 | 6255000 |
| 28 | 兵庫県 | 5484000 |
| 1 | 北海道 | 5286000 |
| 40 | 福岡県 | 5107000 |
| 22 | 静岡県 | 3659000 |
+-----+-----+-----+
10 rows in set (0.01 sec)

mysql> select id, 都道府県, 人口 from example order by 人口 limit 10;
+-----+-----+-----+
| id | 都道府県 | 人口 |
+-----+-----+-----+
| 31 | 鳥取県 | 560000 |
| 32 | 島根県 | 680000 |
| 39 | 高知県 | 706000 |
| 36 | 徳島県 | 736000 |
| 18 | 福井県 | 774000 |
| 19 | 山梨県 | 817000 |
| 41 | 佐賀県 | 819000 |
| 30 | 和歌山県 | 935000 |
| 37 | 香川県 | 962000 |
| 5 | 秋田県 | 981000 |
+-----+-----+-----+
10 rows in set (0.03 sec)

mysql>
```

1.9.5. 簡単な集計

また、簡単な集計程度であればSQLで書ける。例として、人口に対する学生の割合を算出し、高い順に10個のデータを取得する。

```
mysql> select id, 都道府県, 学生数/人口*100 from example order by 学生数/人口*100 desc limit 10;
```

id	都道府県	学生数/人口*100
26	京都府	6.2900
13	東京都	5.4001
27	大阪府	2.6883
17	石川県	2.6103
23	愛知県	2.5436
4	宮城県	2.4303
40	福岡県	2.3513
25	滋賀県	2.3003
28	兵庫県	2.2570
33	岡山県	2.2183

```
10 rows in set (0.00 sec)
```

```
mysql>
```

2. node.jsからmySQLにアクセスする

2.1. パッケージのインストール

node.jsでは、拡張機能をパッケージと呼び、簡単なコマンドでインストールすることができる。そのコマンドはnpm (node.js package manager) であり、Paiza Cloudではすぐに使えるようになっている。それでは、mySQLにつなぐためのパッケージをインストールしよう。

```
~/websystem2$ npm install mysql
+ mysql@2.17.1
added 11 packages from 15 contributors in 1.594s
~/websystem2$
```

2.2. server5の起動

server5.jsを動かすと、データベースに接続してその結果をコンソールに表示する。本来はWebブラウザに返すのであるが、その前段階としてmySQLとの接続及び通信方法に注目して欲しい。

server5.jsの内容は以下の通り。

```
const http = require('http');
const url = require('url');
const server = http.createServer();
```

```
const mysql = require('mysql');

var connection = mysql.createConnection({
  host: 'localhost',
  port: 3306,
  user: 'node',
  password: 'websystem',
  database: 'web'
});

server.on( 'request', function(req,res) {

  connection.connect( function(error) {
    if( error) {
      console.log('Connection Error');
      return;
    }
  });

  let url_parse = url.parse(req.url,true);
  res.writeHead( 200, {'Content-Type' : 'text/html' });
  res.write('<!DOCTYPE html>');
  res.write('<html lang=ja>');
  res.write('<head><meta charset="UTF-8"></head>');
  res.write('<body>');
  res.write('<h1>Hello world</h1>');
  connection.query('select id, 都道府県, 人口 from example order by 人口
desc limit 10;', function(error, rows, fields) {
    if( error ) {
      console.log('Query Error');
    }

    for( let i=0; i<rows.length; i++ ) {
      console.log( "id=" + rows[i].id );
      console.log( "都道府県=" + rows[i]['都道府県'] );
      console.log( "人口=" + rows[i]['人口'] );
    }
  });
  connection.end();
  console.log(url_parse);
  res.write('</body>');
  res.write('</html>');
  res.end();
});

server.listen(80);
```

以下のようにして実行して、Webブラウザで接続してみよう。

```
~/websystem2$ sudo node server5.js
```