

# データベース論・演習

第7回 データベースプログラミング

2021/12/2

# 課題7-1 出身地で検索する

- ex7-1のMain.javaを書き換え、出身地を入力すると、その出身地の人のリストを表示するプログラムを作ろう

変更する部分:

1. ユーザから県名を入力してもらうようにする
2. 検索条件を  
    `birthplace = '県名'`  
    になるようにする
3. 検索結果の表示を変える

# ユーザから県名を入力してもらう

```
12 Scanner scanner = new Scanner(System.in);  
13 System.out.println("5000件の名簿から出身地で検索をするよ。県名を入れてね");  
14 String pref = scanner.next();
```

# 検索条件を変更する

```
16  try{
17      Connection conn = DriverManager.getConnection("jdbc:sqlite:meibo.db");
18
19      String query = "select lastname, firstname, birthday from meibo where
20                      birthday = ? order by birthday";
21
22      int count = 0;
23      PreparedStatement ps = conn.prepareStatement(query);
24      ps.setString(1, pref);
```

# 検索結果表示を変える

```
21  int count = 0;
22  PreparedStatement ps = conn.prepareStatement(query);
23  ps.setString(1,pref);
24  ResultSet rs = ps.executeQuery();
25  while(rs.next()){
26      System.out.println(rs.getString(1)+rs.getString(2)+"(出身地:"+rs.getString
        (3)+")");
27      count++;
28  }
29  System.out.println(pref+"県生まれの人は"+count+"人いました。");
```

# 課題8-1 自分で新しい検索プログラムを作ってみよう

- 自販機データベース(vending\_machine.db)を使って課題5でやったような検索ができるプログラムを作る
  - 例1: 技大天久保の自販機プログラムを作ってみる  
(売り切れてない商品リストを出す→選択したものの商品の値段を言ってお金を入れてもらう→「ありがとうございました！」)
  - 例2: 売上分析プログラムを作る
    - 商品名を入れたら、どの自販機で何個売れたかを出力する
  - 例3: 在庫チェックプログラムを作る
    - 残り個数を入れたら、それ以下の個数しか在庫がない飲み物のリストを出す

```
16 try{
17     Connection conn = DriverManager.getConnection
18     ("jdbc:sqlite:meibo.db");
19
20     String query = "select lastname, firstname, birthday from meibo
21     where birthday like ? order by birthday";
22
23     int count = 0;
24     PreparedStatement ps = conn.prepareStatement(query);
25     ps.setString(1,"%"+year+"%");
26     ResultSet rs = ps.executeQuery();
27     while(rs.next()){
28         System.out.println(rs.getString(1)+rs.getString(2)+"(誕生日:"
29         +rs.getString(3)+")");
30         count++;
31     }
32     System.out.println(year+"年生まれの人は"+count+"人いました。");
33 }catch(Exception e){
34     System.out.println(e.getMessage());
35 }
```

1. データベースに接続する

2. 問合せを設定する

3. 問合せを実行する

4. 回答を取り出す

どの言語、どのライブラリ、どのDBMS相手でも  
基本的にプログラムの流れは一緒。

# グループで作ろう

- グループはあらかじめこちらで決めています
- replitで同じプログラムを観れるので、チャットなどを駆使してやってみよう
- プログラムを書く担当を五分ずつ交代し、周りのみんながどう書くかをチャットとか手話で助言するというやりかたをお勧めします。



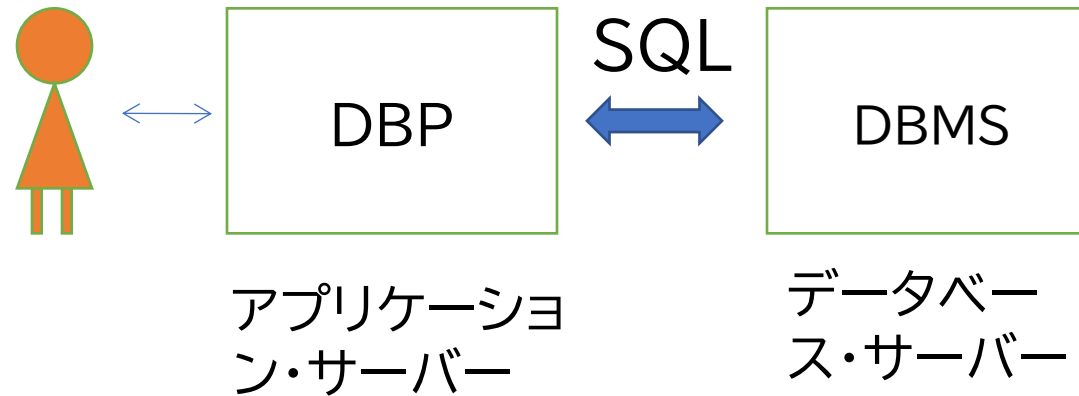
# スケジュール

- 14:50-15:30 プログラムを作る
- 15:30-16:20 内容の発表
- 16:20-16:30 休憩
- 16:30-17:00 講義
- 17:00- 演習

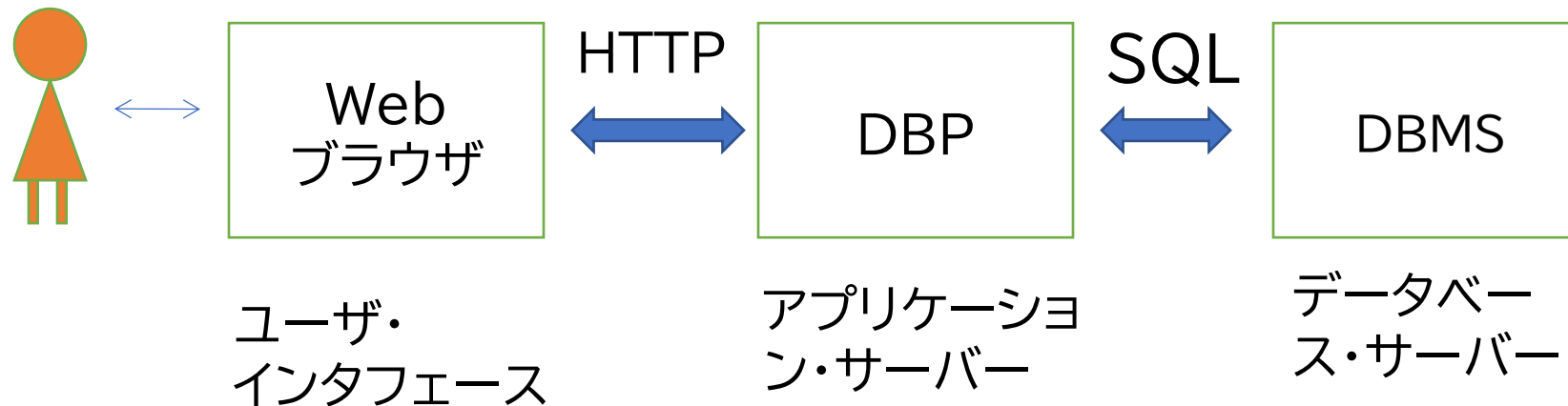
Web+DBプログラミング

# データベースプログラム(略してDBP)の動作形態

## (1) 応用プログラム



## (2) WEB+DB



# 作ってみるプログラム



A screenshot of a web browser window. The address bar shows the URL `https://ex8-2.database2021.repl.co`. The page title is "県名で名簿検索" (Search by Prefecture Name). Below the title, there is a text input field with the placeholder text "県名を入力してください。" (Please enter the prefecture name). To the right of the input field is a button labeled "検索" (Search). Below the input field and button, the text "検索結果" (Search Results) is displayed.

県名で名簿検索

県名を入力してください。

検索

検索結果

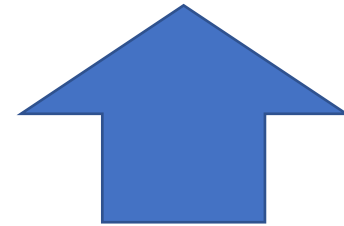
# 結構ボリュームがあるので 段階的にやります

1. Webアプリで、URLを指定してページが表示されるまで
2. データベースを検索する(ユーザの入力なし)
3. フォームでユーザの入力を受け取る
4. 2と3を合わせる

# Step1: WebアプリでURLを指定して ページが表示されるまで

https://**www.abc.de.jp**/**meibo/search**

Webサーバの住所      サーバ内の住所



ルーティング:サーバ内の住所と  
実行するプログラムの対応関係を  
定義すること

# Node.JSのルーティング

サーバ内の住所

request: URLを指定するときと一緒に入力されるパラメータ類が入っている(後述)

response: 実行結果を表示するためのオブジェクト

```
11 app.get('/', function(request, response){
12   const current = new Date();
13   const data = {
14     'month': current.getMonth()+1,
15     'day': current.getDate(),
16     'hour': current.getHours(),
17     'minute': current.getMinutes()
18   };
19   response.render("index.ejs", data);
20 }
21 );
```

実行するプログラム

# 今プログラムで実行していること

```
11 app.get('/', function(request, response){
12     const current = new Date();
13     const data = {
14         'month': current.getMonth()+1,
15         'day':current.getDate(),
16         'hour':current.getHours(),
17         'minute':current.getMinutes()
18     };
19     response.render("index.ejs",data);
20
21 });
```

大事なのはこの1行。Webページを表示する関数

1つ目の引数:

Webページのテンプレート

2つ目の引数:

フォーマットからWebページと作る時に  
埋め込むデータ



# テンプレート

EJS: JavaScriptのテンプレートエンジン。  
テンプレートの中でJavascriptのプログラムを動かしてWebページなどを作る。

- HTMLのBodyの中身
- `<%- month %>`と`<%- day %>`
  - 第2引数で受け取ったデータをテンプレートに埋め込んで最終課題ページを作っている

views/index.ejs ×

```
1   こんにちは!
2   本日は<%- month %>月<%- day %>日ですね。
3   <p>
4   <ul>
5   <li><a href="form_sample">サンプル1:フォームによる値渡し
6   </a></li>
7   <li><a href="query_sample">サンプル2:データベース検索
8   </a></li>
9   </ul>
```

こんにちは！ 本日は12月1日ですね。

- [サンプル1：フォームによる値渡し](#)
- [サンプル2：データベース検索](#)

# Step2: データベースを検索する

```
11  const sqlite3 = require('sqlite3').verbose();
12  const fs = require("fs");
13  const dbfile = "meibo.db";
14  const exists = fs.existsSync(dbfile);
15  const db = new sqlite3.Database(dbfile);
--
```

```
36  app.get('/query_sample', function(request, response){
37      const query = "select firstname, lastname, birthday, birthplace from
                        meibo";
38      db.all(query, function(err, rows){
39          const data = {
40              'results': rows
41          };
42          response.render("query_sample.ejs", data);
43      });
44  });
```

db.all 検索して結果を全部取ってくる

検索が終わったら  
実行するプログラム。  
rowsに検索結果が配列で入っている

## Step2: データベースを検索する

```
38 db.all(query, function(err, rows){
39     const data = {
40         'results': rows
41     };
42     response.render("query_sample.ejs", data);
43 });
```

data.resultsに検索結果を代入して、  
Webページを作る材料にする

# Step2: データベースを検索する

- テンプレートに検索結果を埋め込む

```
2  <ul>
3    <% for(i=0;i<results.length;i++){ %>
4      <li>
5        <%- results[i].lastname %>
6        <%- results[i].firstname %>
7        (<%- results[i].birthday%> ,
8        <%- results[i].birthplace%>)
9      </li>
10   <% } %>
11 </ul>
```

<% %> で挟んだところに  
Javascriptのプログラムをかけるので  
ここでfor文を回している。

results[i].lastname  
検索結果のi番目の行のlastnameの値

# Step2: データベースを検索する

- テンプレートに検索結果を埋め込む

```
2  <ul>
3    <% for(i=0;i<results.length;i++){ %>
4      <li>
5        <%- results[i].lastname %>
6        <%- results[i].firstname %>
7        (<%- results[i].birthday%> ,
8        <%- results[i].birthplace%>)
9      </li>
10   <% } %>
11 </ul>
```

https://FantasticSmugGlitches.chiemi627.repl.co

- 浜崎 祐介 (1981/01/28 , 長野県)
- 井原 紗希 (1967/02/24 , 秋田県)
- 辻 敏昭 (1997/11/27 , 大阪府)
- 篠崎 新吉 (1992/03/08 , 茨城県)
- 大津 治虫 (1985/12/28 , 福岡県)
- 塩田 喜代子 (1976/04/13 , 鳥取県)
- 川井 栄蔵 (1982/11/17 , 宮城県)
- 上田 菜奈 (1990/02/22 , 福井県)
- 菊池 広幸 (1972/10/02 , 岐阜県)
- 石黒 美千子 (1994/03/03 , 山梨県)
- 赤坂 政治 (1979/11/04 , 大分県)
- 大沢 英次 (1985/12/30 , 茨城県)
- 露木 光枝 (1974/09/20 , 沖縄県)

# Step3: フォームでユーザの入力を受け取る

- HTMLで入力フォームをかける

```
1 <form action="form_sample" method="get">
2   <input name="keyword"/>
3   <input type="submit" value="検索">
4 </form>
5
6 入力されたキーワード: <% keyword %>|
```

action = "form\_sample"  
検索ボタンを押したら、  
サーバ内の住所 /form\_sample を実行。

<input name="keyword">  
とあり、keywordという変数名で  
サーバプログラムに渡せる。

入力されたキーワード：



## Step3: フォームでユーザの入力を受け取る

- /form\_sample でフォームに入力された値を受け取る

```
29 app.get('/form_sample', function(request, response){
30     const data = {
31         'keyword': request.query.keyword
32     }
33     response.render("form_sample.ejs", data);
34 });
```

<input name="keyword">で指定されたところ  
に入力された文字は、こちらで使える。  
request.query.keyword

## 課題8-2

- フォームで県名を書いてもらって、その県出身のメンバーの氏名を出力するWebページを作ろう
  - サーバ内の住所は /birthplace
  - プログラム内の変数は好きなように変えて良い
  - ヒント: Step2とStep3を組み合わせる



# 補足: JavascriptのSQLプレースホルダ (?の部分に変数の値を埋め込むこと)

- Javaとは書き方が違います

```
//変数を lastname とし、そこに“渡辺”が入っているとする  
const lastname = “渡辺”;
```

```
const query = “SELECT * FROM meibo WHERE gender = ?”;  
db.all(query, [lastname], function(err,rows){  
  ...  
});
```

?に入るものを配列で指定している