

## データベースS

第10回 PHPによるWebインターフェース(1)

システム創成情報工学科 尾下 真樹

## 今日の内容

- 前回の復習
- 前回の演習の復習
- Web、PHP のしくみ
- HTML+PHP 入門
- PHPによるインターフェース作成(1)

## 参考書

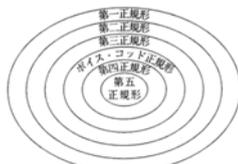
- 「PHP5 徹底攻略」  
堀田 倫英、桑村 潤 著  
ソフトバンクパブリッシング (3,800円)  
– PHP(本日説明) + PostgreSQL  
についての詳しい参考書
- 「PostgreSQLによるLinuxデータベース構築」  
廉升烈 著、翔泳社 出版 (2,200円)



## 前回の復習

## 正規形と正規化

- 正規形の種類
  - 第1正規形
  - 第2正規形
  - 第3正規形
  - ボイス・コード正規形
  - 第4正規形
  - 第5正規形
- 関数従属性や多値従属性に注目して、問題のあるレレーションを段階的に分解していくことで、より厳しい正規形にすることができる
- 通常は第5正規形まで分解することが望ましい



データベースシステム 図4.7

## 関数従属性と多値従属性

- 関数従属性  $X \rightarrow Y$ 
  - 属性(の組)Xが決まれば、属性(の組)Yが一意に決まる
- 多値従属性  $X \twoheadrightarrow Y$ 
  - ある属性(の組)Xについて、いくつかの属性(の組)Yが存在すれば、必ず全てのXY(RS→XY)の組み合わせが存在する
    - RSはリレーションの全ての属性
  - 関数従属性は多値従属性の特殊なものと言える
    - Yが常に1種類のみ存在するもの

### 正規形の条件のまとめ(1)

- 第2正規形
  - 候補キー以外の属性は、候補キーの部分属性に関数従属しない(キー属性が複数属性の組であるときのみ満たさない可能性がある)
- 第3正規形
  - 候補キー以外の属性は、候補キー以外に関数従属しない
- ボイス・コッド正規形
  - 候補キーの部分属性は、非候補キーに関数従属しない(キー属性が複数属性の組であるときのみ満たさない可能性がある)



### 正規形の条件のまとめ(2)

- 第4正規形
  - 多値従属が存在しない(分解後のリレーションが超キーでない)
  - 多値従属性 プロジェクト番号 →→ 社員番号 | ミーティング日
- 第5正規形
  - 結合従属性が存在しない(分解後のリレーションが超キー)
  - 結合従属性 \*([工場番号, 部品番号], [部品番号, 業者番号], [工場番号, 業者番号])



### リレーションスキーマの設計

- DBMSを利用するためには、自分の扱いたい自然界のデータを、DBMSの提供するデータモデルを使って記述する必要がある
  - 概念設計
    - ・ 現実のデータの概念を整理
  - 論理設計
    - ・ 具体的なスキーマの記述を決定



### リレーションスキーマの設計

- 概念設計の方法
  - 実体関連モデル
- 論理設計の方法
  - 実体関連モデルからスキーマを決定
  - 仮のスキーマを正規化していくことで、スキーマを決定

### 実体関連モデル

- 実体関連図(ER図)
  - 実体関連モデルを使ってモデル化した概念を図に表したもの
    - ・ 実体は四角、関連はひし形、属性は丸、キー属性はアンダーラインで表されている



### 正規化による論理設計

- スキーマを作成
  - 履修(学生番号、科目番号、氏名、所属学部、所属学科、住所、科目名、単位数、成績)
- 関数従属性のリストアップ
  - 学生番号 → 氏名、専攻、住所
  - 科目番号 → 科目名、単位数
  - 所属学科 → 所属学部
  - ※ 自明な関数従属性(候補キー全体→他の属性)はリストアップしても、しなくても構わない
    - ・ 例: 学生番号、科目番号 → 成績

## 正規化による論理設計

- 各正規形を満たすように、分解していく
- 分解後のスキーマ
  - 履修(学生番号、科目番号、成績)
  - 学生(学生番号、氏名、所属学科、住所)
  - 学科(所属学科、所属学部)
  - 科目(科目番号、科目名、単位数)

## 前回の演習の復習

## PosgreSQLの使い方

- データベースの作成
- psqlの起動
- テーブルの作成
- データの挿入
- SQLによる問い合わせ
- データの更新と削除
- 複数のテーブルと外部参照整合性制約

## データベース作成

```
username@pcXX ~
# createdb dbname -h popuradb.ces.kyutech.ac.jp -E UTF8
CREATE DATABASE
# psql dbname -h popuradb.ces.kyutech.ac.jp
Welcome to psql 7.3.2, the PostgreSQL interactive terminal.
Type: \copyright for distribution terms
      \h for help with SQL commands
      \? for help on internal slash commands
      \g or terminate with semicolon to execute query
      \q to quit
dbname=#
```

dbname には、データベースの名前を指定

dbname は、必ず自分のアカウント名とすること

## 前回の演習課題

- 資料の説明に従って、データベースを作成し、データを追加

従業員

従業員番号	部門番号	氏名	年齢
0001	01	尾下 真樹	27
0002	02	下戸 彩	17
0003	03	本村 拓哉	30
0004	01	宇田 ヒカル	20
0005	01	織口 裕二	20

部門

部門番号	部門名
01	開発
02	営業
03	総務

- SQLを使った問い合わせ

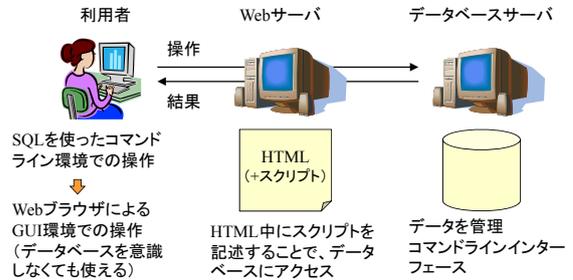
## Web、PHP のしくみ

## データベースとWebインターフェース

- PostgreSQLデータベース(前回まで)
  - psql によるコマンドラインからの操作
  - 操作が面倒
- Webインターフェースの追加(今回の演習)
  - データベースをWebインターフェースから操作
    - データの挿入・修正・削除などの管理
    - 検索結果の表示
  - ユーザはデータベースを意識する必要はない
    - 実際に多くのウェブページの裏ではデータベースが動いている(ショッピング、各種予約など)

## Webインターフェース

- Webページを経由してデータベースを操作



## WWWの仕組み

- クライアントの要求に応じて、WebサーバがHTMLファイルを返す
  - URLによる表示対象(ファイル)の指定
    - 例: <http://www.cg.ces.kyutech.ac.jp/~oshita/index.html>
      - プロトコル      サーバ名      ファイル名
  - HTML(Hyper-Text Markup Language)
    - ページの内容やレイアウトをテキストファイルで記述
    - ブラウザはHTMLを解釈してページを表示
    - 単純なHTMLだけでは、あらかじめ作成されたページしか表示できない
      - 掲示板のように内容が変化するページは実現できない

## HTMLファイルの例

- メニュー(menu.html)

```
<HTML>
<HEAD>
  <TITLE>データ操作メニュー</TITLE>
</HEAD>
<BODY>
  操作メニュー<BR>
  <UL>
    <LI><A HREF="employee_list.php">従業員の一覧表示</A>
    <LI><A HREF="employee_add_form.html">従業員のデータ追加</A>
    <LI><A HREF="employee_add_form.php">従業員のデータ追加(動的生成版)</A>
    <LI><A HREF="employee_delete_form.html">従業員のデータ削除</A>
    <LI><A HREF="employee_delete_form.php">従業員のデータ削除(動的生成版)</A>
    <LI><A HREF="employee_update_form1.html">従業員のデータ更新</A>
  </UL>
</BODY>
</HTML>
```

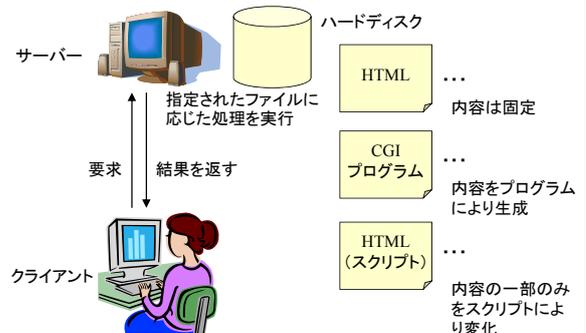
## HTMLファイルの表示結果の例

- ウェブブラウザでの表示結果
  - フォントの種類や大きさ等は、ブラウザの設定により異なる

操作メニュー

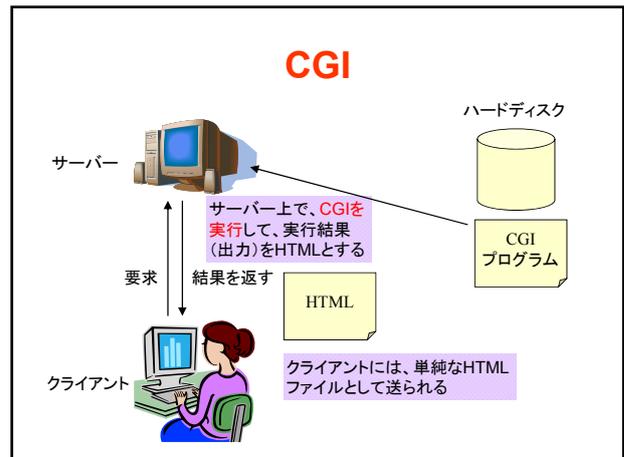
- [従業員の一覧表示](#)
- [従業員のデータ追加](#)
- [従業員のデータ追加\(動的生成版\)](#)
- [従業員のデータ削除](#)
- [従業員のデータ削除\(動的生成版\)](#)
- [従業員のデータ更新](#)

## WWWの仕組み



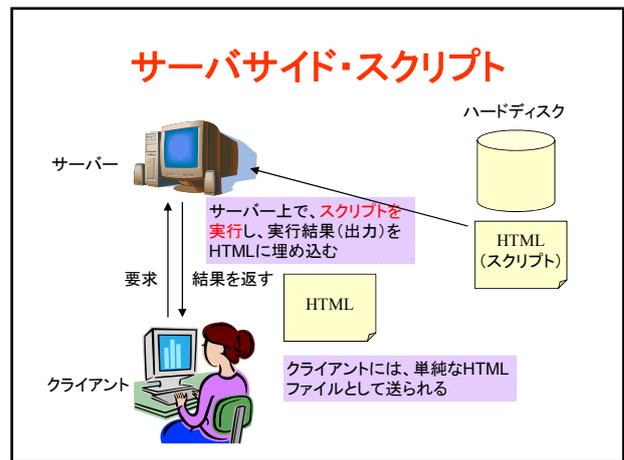
## CGIとスクリプト言語

- CGI
  - Perl や C/C++ などのプログラミング言語を使って動的にHTMLを生成する技術
  - HTMLファイルの代わりに、プログラム名をURLで指定し、プログラムの出力したテキストを送信
- スクリプト
  - HTMLの中にプログラムを記述しておき、そのプログラムによってHTMLを動的に変化させる
    - ・ スクリプト=プログラム(比較的簡単な物をスクリプトと呼ぶ)
  - サーバサイド・スクリプトとクライアントサイド・スクリプトの2種類がある



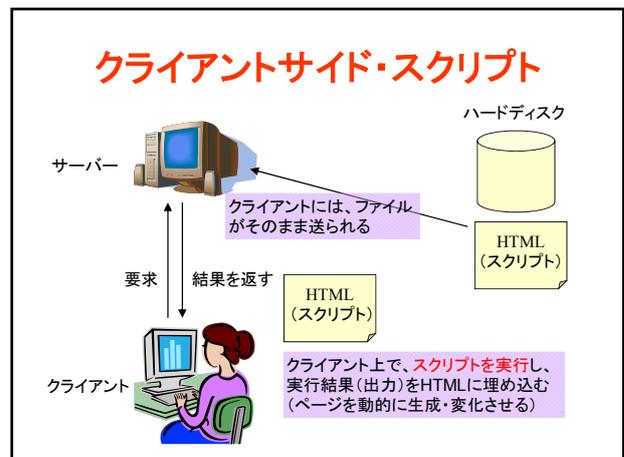
## サーバサイド・スクリプト

- サーバ側で動作するスクリプト
  - PHP や SSI など
  - サーバ側で実行されて、HTMLテキストとしてクライアント側に送られる
    - ・ サーバの機能を使用できるので、データベース処理などの高度な処理を行うのに適している
    - ・ CGIと同様に、クライアント側にはプログラムは送られないため、利用者にプログラムを見られる心配がない



## クライアントサイド・スクリプト

- クライアント側のブラウザ上で動作するスクリプト
  - JavaScript や VBScript など
  - HTML中に含まれたままクライアントに送られ、ブラウザ上で実行される
    - ・ HTMLが表示された後も実行し続けることができるため、アニメーションや対話的操作を含む機能の実現に適している
  - 最近では、HTML 5 や AJAX (ブラウザ上での対話的な操作を実現)の要素技術として、盛んに利用されている
    - ・ Google Map, Gmail など
  - Java や Flash もクライアント側で実行されるという点は同じだが、HTMLとは別のJavaやFlashのプログラムが実行・表示されるという点でやや異なる



## 機能の比較

- CGI
  - 全てをプログラムで出力する必要があるので、ウェブページに固定の部分と動的に生成される部分が混在していると、固定部分の管理が面倒
    - プログラムに全て埋め込み or 別ファイルから読み込み、など
- サーバサイドスクリプト
  - HTMLの一部に、スクリプトの出力が埋め込まれるので、ウェブページの一部のみを動的に生成するのに適している
    - 固定部分と動的に生成される部分が同一ファイルで管理される

## PHP

- サーバサイド・スクリプトの一種
  - サーバ側で働くスクリプト
  - HTMLとPHPスクリプトの混在したソースを記述
  - サーバ側でPHPスクリプトを実行
    - PHPスクリプトから出力したテキストが、HTMLに追加される
  - ブラウザには、最終的なHTMLが送られる
- 本演習では、PHPを使って、Webインターフェースを作成する

## HTML + PHP 入門

## HTMLの基礎

- テキスト+タグ
  - タグで囲むことによって、テキストの属性を指定する
    - 例: `<B>`太字になります`</B>`
  - ハイパーリンク(他のページへのリンク)などが記述できる
  - 基本的なタグ
    - リンク、改行、テーブル、箇条書き
    - 画像などのタグについて知りたい人は、各自、適当な資料で勉強してください

## HTMLの構成

```
<HTML>
<HEAD>
  ここには、ページに関する情報を記述
  <TITLE>ページのタイトル</TITLE>
</HEAD>
<BODY>
  ここに本文を書く。
</BODY>
</HTML>
```

## HTMLファイルの例

- メニュー(menu.html)

```
<HTML>
<HEAD>
  <TITLE>データ操作メニュー</TITLE>
</HEAD>
<BODY>
  操作メニュー<BR>
  <UL>
    <LI><A HREF="employee_list.php">従業員の一覧表示</A>
    <LI><A HREF="employee_add_form.html">従業員のデータ追加</A>
    <LI><A HREF="employee_add_form.php">従業員のデータ追加(動的生成版)</A>
    <LI><A HREF="employee_delete_form.html">従業員のデータ削除</A>
    <LI><A HREF="employee_delete_form.php">従業員のデータ削除(動的生成版)</A>
    <LI><A HREF="employee_update_form1.html">従業員のデータ更新</A>
  </UL>
</BODY>
</HTML>
```

ヘッダ情報

本文

## HTMLファイルの表示結果の例

- ウェブブラウザでの表示結果
  - フォントの種類や大きさ等は、ブラウザの設定により異なる

操作メニュー

- [従業員の一覧表示](#)
- [従業員データの追加](#)
- [従業員データの追加\(動的生成版\)](#)
- [従業員データの削除](#)
- [従業員データの削除\(動的生成版\)](#)
- [従業員データの更新](#)

## HTMLの基本的なタグ

- `<BR>` 改行
- `<HR>` 水平線
- `<A>` 他のページへのリンク
  - `<A HREF="http://www.ces.kyutech.ac.jp">学科のページへ</A>`
  - `<A HREF="menu.html">同一ディレクトリにある別のページへ</A>`
  - `<A HREF="sub/test.html">サブディレクトリにあるページへ</A>`
  - `<A HREF="./index.html">親ディレクトリにあるページへ</A>`
- `<TABLE>`, `<TR>`, `<TD>` テーブル
- `<!-- コメント -->`

## フォーム

- ウェブページに入力できる仕組み
  - `<FORM>` ~ `</FORM>`
  - 送信ボタンを押すと、指定したURLを呼び出し
  - フォーム記入したデータをURLで指定したプログラムに引数として送信できる
    - データの受け渡し方に、GETとPOSTの2種類がある
  - 詳しくは、後で使用方法も含めて説明

従業員データ追加フォーム

従業員番号:  部門番号:  氏名:  年齢:  性別:  男  女

## PHPの記述(1)

- HTML内へのPHPスクリプトの記述
  - `<?php ~ ?>`
- PHPスクリプト
  - if や while などの制御構文は、Java や C と同じ
- 変数
  - \$で始まる文字列を変数とみなす
  - 宣言せずに使って良い
  - 型は指定しなくても良い(値により自動的に決まる)
  - JavaやCとは、上記の点が大きく異なるので注意

## PHPスクリプトを含むHTMLの例

```
<HTML>
<HEAD>
<TITLE>従業員リスト</TITLE>
</HEAD>
<BODY>
<CENTER>
検索結果を表示します。<BR><BR>
<!-- ここからPHPのスタート始まり -->
<?php
// データベースに接続
// ※ your_db_name のところは自分のデータベース名に書き換える
$conn = pg_connect("dbname=your_db_name");
// 接続が成功したかどうか確認
if ( $conn == null )
{
    print("データベース接続処理でエラーが発生しました。<BR>");
    exit;
}
```

PHPスクリプトの開始

## PHPの記述(2)

- 演算子
  - 基本的には、Javaと同じ(+ \* / && || など)
  - 文字列の結合には「.」を使う
    - + を使うと、自動的に数値型に型変換してから、数値型として足し算が計算されてしまうので注意

```
// 変数xには、文字列型の "12345678" が入る
$x = "1234" . "5678";
```

```
// 変数xには、整数型の 6912 が入る
$x = "1234" + "5678";
```

### PHPの記述(3)

- テキスト出力
  - PHPスクリプト中で文字列を出力すると、HTMLに書き出される
    - ページの内容を動的に生成できる
  - print( 文字列 );
    - 文字列の出力 (文字列中に変数名を書くことで、変数値を文字列に直接埋め込むことができる)
  - print( 書式付文字列, 値1, 値2, ... );
    - 文字列の一部に変数の値などを埋め込める
  - sprintf(書式付文字列, 値1, 値2, ... );
    - printfと同様の出力結果を文字列として返す

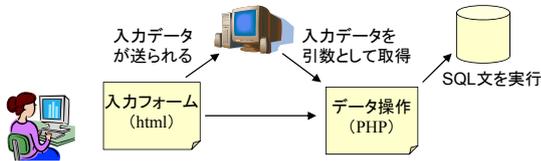
### PHPからPostgreSQLの操作

- 専用の関数が用意されている
  - pg\_ で始まる関数
    - データベースに接続
  - pg\_connect( string option );
    - クエリーを実行
  - pg\_query( query );
    - クエリーの結果の行数を取得
  - pg\_num\_rows( result );
    - クエリーの結果のテーブルから i行j列の値を取得
      - ij は0 から始まることに注意(例:2行3列目 → i=1, j=2)
  - pg\_fetch\_result( result, i, j );
    - pg\_close();

従業員番号	部門番号	氏名	年齢
0001	01	尾下 真樹	27
0002	02	下戸 彰	17
0003	03	本村 拓哉	30
0004	01	宇田 仁カル	20
0005	01	織口 裕二	20
0006	02	松浦 幸矢	20
0007	03	山田 一郎	20

### PHPでの引数の受け取り

- スーパーグローバル変数経由で取得
  - フォーム側が GET で出力した場合
    - \$\_GET[ 引数名 ]
  - フォーム側が POST で出力した場合
    - \$\_POST[ 引数名 ]



### 引数の受け渡し方の違い

- GET
  - URLに付与する形で受け渡し(利用者に引数が見える)
  - サーバー側のプログラムは、環境変数という仕組みを使って受け取る
- POST
  - ウェブブラウザとサーバーが通信をして受け渡し(利用者には引数が見えない)
- 両者の使い分け
  - 一般に、POSTの方が処理が面倒な代わりに、高機能
  - PHPで使う場合は、PHPが細かい処理をやってくれるので、どちらも簡単に使える
  - 本演習では、引数が分かりやすいように、GETを使用

### SQL文の作成

- SQL文は文字列として扱える
  - \$sql = "select \* from employee where id='001'";
    - 注意: " (ダブルクォート)はPHPの文字列の区切り、' (シングルクォート)はSQLの文字列の区切り
- 文字列を埋め込むことで動的にSQL文を作成できる (以下の3つは、どれも同じ結果になる)
  - \$sql = "select \* from employee where id=" . \$id . "";
  - 文字列の連結
  - \$sql = "select \* from employee where id='\$id'";
  - \$id の値が文字列に埋め込まれる
  - \$sql = sprintf("select \* from employee where id=%s", \$id );
  - 文字列中の %s の箇所が、\$id の値で置き換えられる

### PHPによるインターフェース作成

## Webインターフェースの作成

- 従業員・部門のデータベースの操作
  - 一覧表示、追加、削除、更新
- サンプルのHTML・PHPスクリプト
  - Moodleの本講義のページに置いてある
  - 各自ダウンロード、適宜修正して、実行してみる

## データベースの準備

- 前回の演習で作成したデータベースを使用
  - 前回の演習を完了していれば、そのままが良い
  - 万一、前回の演習が完了していなければ、前回の演習の資料に従って、データベースを作成
- テーブルの利用権限の設定
  - ウェブサーバのプロセスを実行するシステムユーザ `www` に、テーブルを読み書きする権限を与える (`psql` の `grant` コマンドを使用)
    - ※ ユーザ名は、サーバの設定により異なる
  - 各テーブルにつき一度だけ行えば良い

## ターミナルでの操作

```
username@pcXX ~
# psql dbname -h popuradb.ces.kyutech.ac.jp
Welcome to psql 7.3.2, the PostgreSQL interactive terminal.
. . .

dbname=# grant ALL on employee to www;
GRANT
dbname=# grant ALL on department to www;
GRANT
```

`dbname` には、必ず自分のアカウント名を入れること！

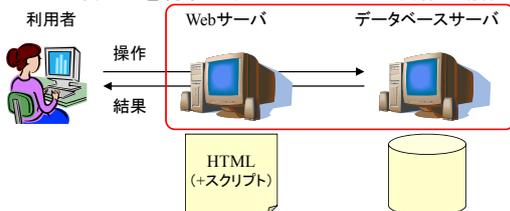
## ウェブページの準備

- ウェブサーバ
  - <http://popuradb.ces.kyutech.ac.jp>
  - 今回はデータベースサーバと同じコンピュータ
  - ※ 学科外からはアクセスできないので注意
- 以下のディレクトリにファイルを置く
  - ホームディレクトリ(¥Z)の `public_html`
- 以下のURLでアクセスできる
  - <http://popuradb.ces.kyutech.ac.jp/~ユーザ名/ファイル名>

## Webサーバ

- 今回は、データベースサーバとWebサーバは同一のコンピュータ

– 同一コンピュータ上にある、サーバ同士が通信して、処理を行う (別のコンピュータでも同様に動作)



## 演習手順

- データベースの準備
  - テーブルの作成、データの追加 (前回終了)
  - テーブルの利用権限の設定
- html(PHP) ファイルの作成
  1. 講義のページからダウンロードした `menu.html` を適切な場所に置き、表示されることを確認
  2. 同じく `employee_list.php` を置き (一部修正が必要)、従業員一覧が表示されることを確認
  3. 他のファイル (追加、更新、削除) についても、動作を確認 (次回行う)

## インターフェースの作成

- 作成する機能
  - 従業員データの一覧表示
  - 従業員データの追加
  - 従業員データの追加(動的生成)
  - 従業員データの削除
  - 従業員データの削除(動的生成)
  - 従業員データの更新

## サンプルページの構成

- **メニュー(menu.html)**
  - 一覧表示(employee\_list.php)
  - 追加フォーム(exemployee\_add.html)
    - 追加処理(employee\_add.php)
  - 追加フォーム(動的生成版)(exemployee\_add\_form.php)
    - 追加処理(employee\_add.php)
  - 削除フォーム(employee\_delete.html)
    - 削除処理(employee\_delete.php)
  - 削除フォーム(動的生成版)(employee\_delete\_form.php)
    - 削除処理(employee\_delete.php)
  - 更新フォーム(employee\_update\_form1.html)
    - 更新フォーム(employee\_form2.php)
    - 更新処理(employee\_update.php)

## メニュー

- **メニュー(menu.html)**
  - <HTML> <HEAD> <TITLE> <BODY>
  - <UL> ~ </UL> <LI> によるリスト
  - 各機能のページへのリンク
    - <A HREF="..."> ~ </A>

## メニュー

- **メニュー(menu.html)**

```

<HTML>
<HEAD>
<TITLE>データ操作メニュー</TITLE>
</HEAD>
<BODY>
操作メニュー<BR>
<UL>
<LI><A HREF="employee_list.php">従業員の一覧表示</A>
<LI><A HREF="employee_add_form.html">従業員のデータ追加</A>
<LI><A HREF="employee_add_form.php">従業員のデータ追加(動的生成版)</A>
<LI><A HREF="employee_delete_form.html">従業員のデータ削除</A>
<LI><A HREF="employee_delete_form.php">従業員のデータ削除(動的生成版)</A>
<LI><A HREF="employee_update_form1.html">従業員のデータ更新</A>
</UL>
</BODY>
</HTML>
    
```

## 表示結果

- **ウェブブラウザでの表示結果**
  - フォントの種類や大きさ等は、ブラウザの設定により異なる

操作メニュー

- [従業員の一覧表示](#)
- [従業員のデータ追加](#)
- [従業員のデータ追加\(動的生成版\)](#)
- [従業員のデータ削除](#)
- [従業員のデータ削除\(動的生成版\)](#)
- [従業員のデータ更新](#)

## サンプルページの構成

- **メニュー(menu.html)**
  - 一覧表示(employee\_list.php)
  - 追加フォーム(exemployee\_add.html)
    - 追加処理(employee\_add.php)
  - 追加フォーム(動的生成版)(exemployee\_add\_form.php)
    - 追加処理(employee\_add.php)
  - 削除フォーム(employee\_delete.html)
    - 削除処理(employee\_delete.php)
  - 削除フォーム(動的生成版)(employee\_delete\_form.php)
    - 削除処理(employee\_delete.php)
  - 更新フォーム(employee\_update\_form1.html)
    - 更新フォーム(employee\_form2.php)
    - 更新処理(employee\_update.php)

### 一覧表示(1)

- 一覧表示 (employee\_list.php)
  - PHPプログラムの開始 (12行目)
  - データベースへの接続 (16行目)
    - データベース名を、各自の名前に変更する必要がある (前回の資料の通りに作業していれば、自分のアカウント名でデータベースを作成しているはず)
    - 接続情報を \$conn に記録

### 一覧表示(2)

```

<HTML>
<HEAD>
<TITLE>従業員リスト</TITLE>
</HEAD>
<BODY>
<CENTER>
検索結果を表示します。<BR><BR>
<!-- ここからPHPのスク립ト始まり -->
<?php
// データベースに接続
// ※ your_db_name のところは自分のデータベース名に書き換える
$conn = pg_connect( "dbname=your_db_name" );
// 接続が成功したかどうか確認
if ( $conn == null )
{
    print( "データベース接続処理でエラーが発生しました。<BR>" );
    exit;
}
    
```

PHPスクリプトの開始

PostgreSQLデータベースへの接続を行う、PHPの関数

データベース名を指定 (自分のデータベース名に書き換える)

接続情報が返される(失敗時はnull)

### 一覧表示(3)

- 一覧表示 (employee\_list.php)
  - SQL文を実行 (26, 29行目)
    - 全従業員のデータを取得するSQL文 (変数 \$sql)
    - 検索結果のテーブルが \$result に格納される

```

// SQLを作成
$sql = "select id, department_name, employee.name, age from employee,
department where employee.dept_no = department.dept_no order by id";
// Queryを実行して検索結果をresultに格納
$result = pg_exec( $conn, $sql );
if ( $result == null )
{
    print( "クエリー実行処理でエラーが発生しました。<BR>" );
    exit;
}
    
```

### 一覧表示(4)

- 一覧表示 (employee\_list.php)
  - 検索結果の行数・列数を取得 (37, 38行目)
    - SQL文で4つの出力属性を指定しているため、列数は必ず4になる (今回は、わざわざ列数を取得しなくても分かっているが、例のために、取得している)

```

// 検索結果の行数・列数を取得
$rows = pg_num_rows( $result );
$cols = pg_num_fields( $result );
    
```

引数には、さきほどのSQLの実行結果を格納した変数を指定

SQLの実行結果から、行数(データ数)と列数(属性数)を取得するPHPの関数

### 一覧表示(5)

- 一覧表示 (employee\_list.php)
  - テーブルを使って結果を表示 (42~69行目)
    - <TABLE> <TR> <TD>
  - 各データ(検索結果の各行)の情報を順番に表示 (53~65行目)
    - for文を使って、各行・列ごとに繰り返し
  - 検索結果から属性値を取得して表示 (59行目)
    - pg\_fetch\_result( 結果, 行番号, 列番号 )

### 一覧表示(6)

```

// 検索結果をテーブルとして表示
print( "<TABLE BORDER=1>\n" );
// 各列の名前を表示
print( "<TR>" );
print( "<TH>従業員番号</TH>" );
print( "<TH>部門</TH>" );
print( "<TH>氏名</TH>" );
print( "<TH>年齢</TH>" );
print( "</TR>\n" );
.....
    
```

テーブルの開始

1行目の見出しの表示

従業員番号	部門	氏名	年齢
0001	開発	山下真樹	27
0002	営業	下戸彩	17
0003	総務	本村拓哉	30
0004	開発	宇田ヒカル	20
0005	開発	堀口裕二	35
0006	営業	松浦幸矢	17
0007	総務	山田 一郎	30

以上、7件のデータを表示しました。

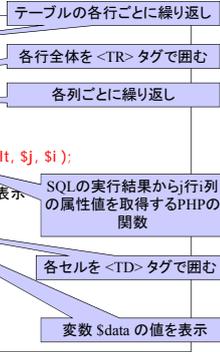
操作メニューに戻る

表示されるテーブル

### 一覧表示(7)

```

// 各行のデータを表示
for ( $j=0; $j<$rows; $j++)
{
    print("<TR>");
    for ( $i=0; $i<$cols; $i++)
    {
        // j行列のデータを取得
        $data = pg_fetch_result( $result, $j, $i );
        // テーブルのj行列に属性値を表示
        print( "<TD> $data </TD>" );
    }
    print( "</TR>\n" );
}
// ここまででテーブル終了
print( "</TABLE>" );
print( "<BR>\n" );
    
```



### 一覧表示(8)

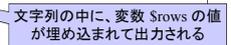
- 一覧表示 (exemployee\_list.php)
  - データ数を表示 (74行目)
  - 終了処理 (78, 81行目)
    - 検索結果の開放
    - データベースへの接続を解除

```

// 検索件数を表示
print( "以上、$rows 件のデータを表示しました。<BR>\n" );

// 検索結果の開放
pg_free_result( $result );

// データベースへの接続を解除
pg_close( $conn );
    
```



### 実行結果の例

```

<HTML>
<HEAD>
<TITLE>従業員リスト</TITLE>
</HEAD>
<BODY>
<CENTER>
検索結果を表示します。<BR><BR>
<!-- ここからPHPのスク립ト始まり -->
<TABLE BORDER="1">
<TR><TH>従業員番号</TH><TH>部門</TH><TH>氏名</TH><TH>年齢</TH></TR>
<TR><TD> 0001 </TD><TD> 開発 </TD><TD> 尾下 真樹 </TD><TD> 27 </TD></TR>
<TR><TD> 0002 </TD><TD> 営業 </TD><TD> 下戸 彩 </TD><TD> 17 </TD></TR>
<TR><TD> 0003 </TD><TD> 総務 </TD><TD> 本村 拓哉 </TD><TD> 30 </TD></TR>
.....
</TABLE><BR>
以上、7 件のデータを表示しました。<BR>
<!-- ここまでPHPのスク립ト終わり -->
<BR>
<A HREF="menu.html">操作メニューに戻る</A>
</CENTER>
    
```



### 表示結果の例

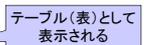
- ウェブブラウザでの表示結果

検索結果を表示します。

従業員番号	部門	氏名	年齢
0001	開発	尾下 真樹	27
0002	営業	下戸 彩	17
0003	総務	本村 拓哉	30
0004	開発	宇田 ヒカル	20
0005	開発	堀口 裕二	35
0006	営業	松浦 亜矢	17
0007	総務	山田 一郎	30

以上、7 件のデータを表示しました。

[操作メニューに戻る](#)



### 演習課題

- 前回の演習課題は終わっており、各自のデータベースは作成されているものとする
- メニュー・一覧表示 (menu.html, employee\_list.php) のファイルをアップロードし、動作確認をする
  - employee\_list.phpは、一箇所、修正が必要
- 一覧表示を行なうPHPプログラムを変更し、従業員の一覧が、部門ごとに表示されるようにする (exemployee\_list.phpを変更)
- 提出締め切り 6月23日(火) 18:00

### まとめ

- 前回の復習
- 前回の演習の復習
- Web、PHP のしくみ
- HTML+PHP 入門
- PHPによるインターフェース作成(1)
  - データの一覧表示
- 演習課題

## 次回予告

- PHPによるインターフェース作成(2)
  - データの更新(追加・削除・修正)
- 演習課題
- 期末レポート課題の説明