

データベースS

第10回 PHPによるWebインターフェース
開発演習 (1)

システム創成情報工学科 尾下 真樹
2018年度 Q2

今日の内容

- 前回の復習
- 前回の演習の復習
- WWWの仕組み
- HTML+PHP の基礎
- 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

前回の復習

正規形

- 更新時に不整合が発生しないような、整合性を保つリレーションスキーマの条件を定義

- 正規形の種類

- 第1正規形
- 第2正規形
- 第3正規形
- ボイス・コッド正規形
- 第4正規形
- 第5正規形



第1正規形→第5正規形まで、徐々に条件が厳しくなっていく

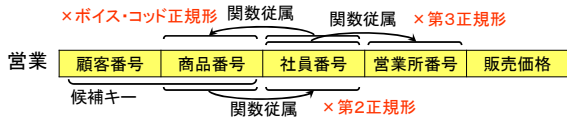
各正規形は、それよりも上の全ての正規形の条件を満たす

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

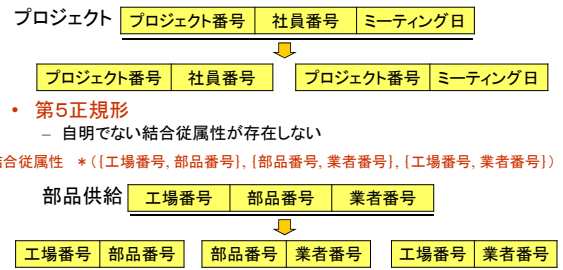
正規形の条件のまとめ(1)

- 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・コード正規形
 - 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)



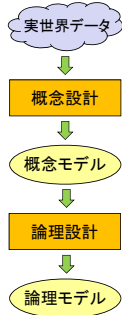
正規形の条件のまとめ(2)

- 第4正規形
 - 自明でない多値従属が存在しない
 - 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
 - 自明でない結合従属性が存在しない
 - 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

- データベースシステムを利用するためには、データベースに格納したい実現データの、データベースシステムが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
 - 実現データの概念を整理
 - 論理設計
 - 具体的なスキーマを決定

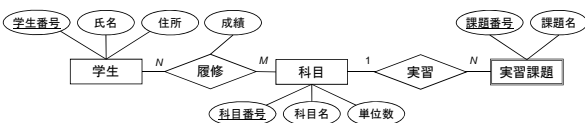


リレーションスキーマの設計

- 概念設計の方法
 - 実体関連モデルを用いる方法
- 論理設計の方法
 - 実体関連モデルからスキーマを決定する方法
 - 仮のスキーマを正規化していく方法

実体関連モデルの記述方法(1)

- 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

PostgreSQL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

前回までの演習課題

- 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 | 織田 信長 | 48 |
| 0002 | 02 | 豊臣 秀吉 | 45 |
| 0003 | 03 | 徳川 家康 | 39 |
| 0004 | 01 | 柴田 勝家 | 60 |
| 0005 | 01 | 伊達 政宗 | 15 |

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

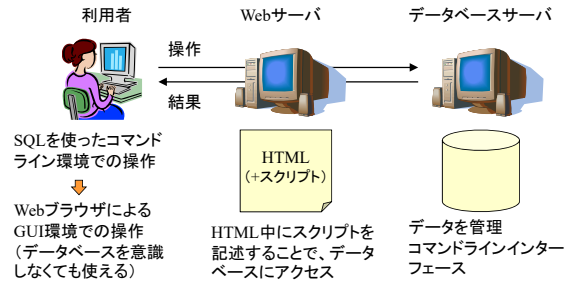
WWWの仕組み

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
 - SQLの知識が必要、操作に手間がかかる
- 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)

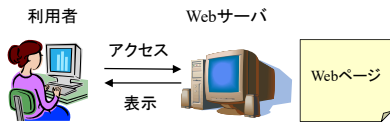
Webインターフェース

- Webページを経由してデータベースを操作



World Wide Web

- World Wide Web (WWW)
 - インターネット上で、Webサーバにアクセスして、相互参照されたドキュメント(Webページ)を閲覧できるようにする技術
 - リンク(参照)を辿ることで多数のページを閲覧できる
 - 一般的に広く利用されている



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固定の内容を表示することしかできない
 - 内容が動的に変化するページの実現には別の技術が必要

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">従業員データの追加</A>
<LI><A HREF="employee_add_form.php">従業員データの追加(動的生成版)</A>
<LI><A HREF="employee_delete_form.html">従業員データの削除</A>
<LI><A HREF="employee_delete_form.php">従業員データの削除(動的生成版)</A>
<LI><A HREF="employee_update_form1.html">従業員データの更新</A>
</UL>
</BODY>
</HTML>
```

HTMLファイルの表示結果の例

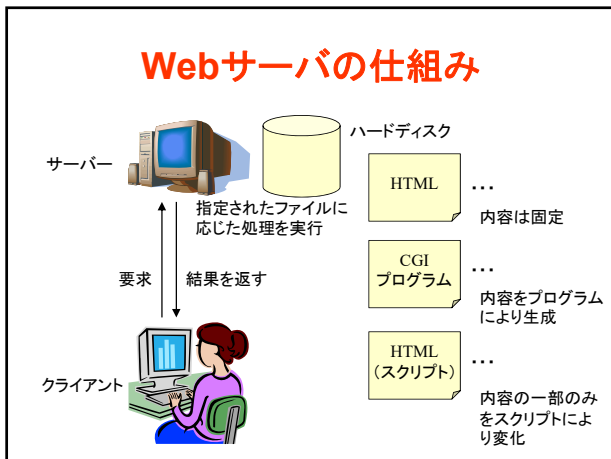
- ウェブブラウザでの表示結果

- 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](#)

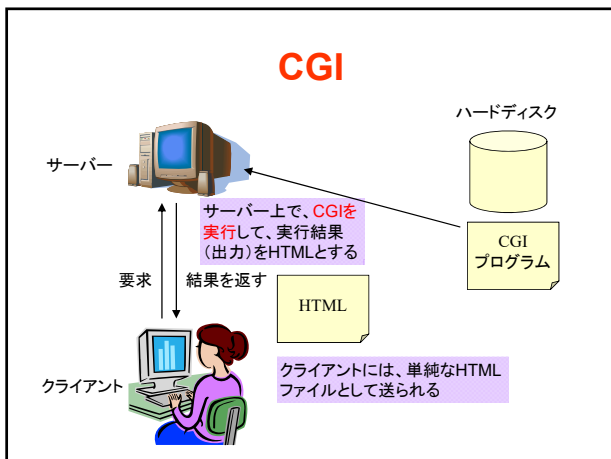
Webサーバの仕組み



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

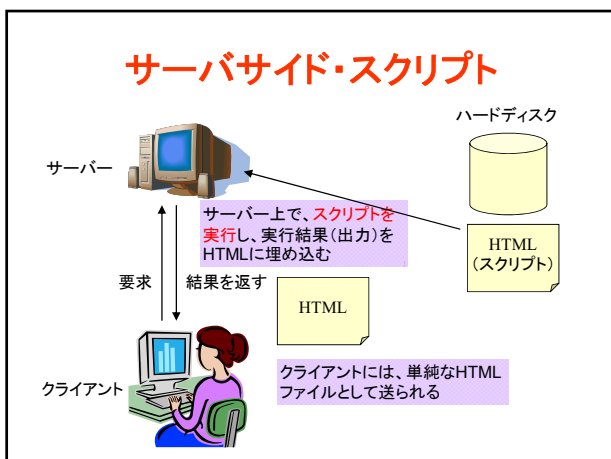
CGI



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

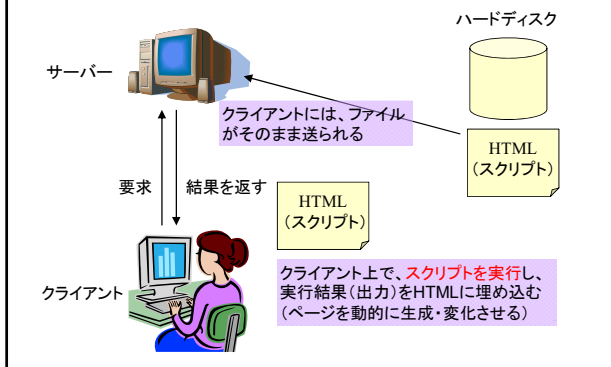
サーバサイド・スクリプト



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

クライアントサイド・スクリプト



機能の比較

- サーバー側で処理を行う技術の比較
- CGI
 - 全てをプログラムで出力する必要があるため、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
 - プログラムに固定部分を記述 or 別ファイルから読み込
- 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ため、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
 - 固定部分と動的生成部分を同一ファイルに記述

PHPの利用

- PHP
 - サーバサイド・スクリプトとしての利用に適したプログラミング言語
 - JavaやC++と同様のオブジェクト指向言語
 - Webシステムの開発に便利な標準関数を備える
 - 多くのWebシステムで使われている (WordPress等)
- 本演習では、PHPを使って、Webインターフェースを作成する

PHPの利用形態

- サーバサイド・スクリプトとしての利用
 - HTML と PHPスクリプト が混在したソースファイルを記述
 - ウェブサーバ側で PHPスクリプトを実行
 - PHPスクリプトから出力されたテキストが、元のHTMLに埋め込まれる
 - クライアント(ウェブブラウザ)には、最終的に生成された HTML が送られる

Webインターフェースの実現

- Webベースのシステムの構成要素
 - オペレーティングシステム (Linux 等)
 - ウェブサーバ (Apache 等)
 - データベースシステム (PostgreSQL, MySQL 等)
 - サーバサイドスクリプト (PHP, Python, Perl 等)
- フリーウェアのみで構築可能
 - 頭文字を取って LAMP, LAPP などと呼ばれる
- Webベースのシステムを構築する上での基礎技術

HTML + PHP の基礎

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: ``太字になります``
 - 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

HTMLの基本的なタグ(1)

- `
` 改行
- `<HR>` 水平線
- `<A>` 他のページへのリンク
 - ``学科のページへ``
 - ``同一ディレクトリにある別のページへ``
 - ``サブディレクトリにあるページへ``
 - ``親ディレクトリにあるページへ``
- `<!-- コメント -->`

HTMLの基本的なタグ(2)

- `<TABLE>`, `<TR>`, `<TD>` テーブル
 - テーブル全体を `<TABLE>` タグで囲む
 - テーブル内の各行を `<TR>` タグで囲む
 - 各行内の各セルを `<TD>` タグで囲む
 - 強調したい項目(見出しなど)は `<TH>` タグで囲む
 - `<TABLE>`内に行数分の`<TR>`、`<TR>`内に列数分の`<TD>`のように、入れ子の形で記述

| 従業員番号 | 部門 | 氏名 | 年齢 |
|-------|----|-------|----|
| 0001 | 開発 | 福田 信典 | 48 |
| 0002 | 営業 | 豊原 秀吉 | 45 |
| 0003 | 総務 | 徳川 孝康 | 39 |
| 0004 | 開発 | 柴田 勝家 | 60 |
| 0005 | 営業 | 伊達 政宗 | 15 |
| 0006 | 総務 | 上杉 景勝 | 26 |
| 0007 | 開発 | 島津 家久 | 35 |

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のスクリプト始まり -->
    <?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)
 - printf(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, ...);
 - printと同様の出力結果を文字列として返す

PHPからPostgreSQLの操作

- 専用の関数を用意されている
 - pg_ で始まる関数
 - pg_connect(string option);
 - データベースに接続
 - pg_query(query);
 - クエリーを実行
 - pg_num_rows(result);
 - クエリーの結果の行数を取得
 - pg_fetch_result(result, i, j);
 - クエリーの結果のテーブルから i行j列の値を取得
 - i, j は 0 から始まることに注意(例:2行3列目→i=1, j=2)
 - pg_close();

| 従業員番号 | 部門番号 | 氏名 | 年齢 |
|-------|------|-------|----|
| 0001 | 01 | 織田 信長 | 48 |
| 0002 | 02 | 豊田 秀吉 | 45 |
| 0003 | 03 | 徳川 家康 | 39 |
| 0004 | 01 | 柴田 勝家 | 60 |
| 0005 | 01 | 伊達 政宗 | 15 |
| 0006 | 02 | 上杉 素勝 | 26 |
| 0007 | 03 | 島本 家久 | 35 |

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開発

Webインターフェースの開発

- 従業員・部門のデータベースの操作を行うことができる Webインターフェースを開発する
 - 一覧表示、追加、削除、更新
- HTML・PHPファイルの作成
 - 元になるサンプルファイルを、Moodle の本講義のページで公開している
 - 各自ダウンロード、適宜修正して、動作確認する
 - 今後の演習で、追加修正を加えていくことで、Webインターフェースを開発する

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 apache 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 apache;  
GRANT  
dbname=# grant ALL on department to apache;  
GRANT
```

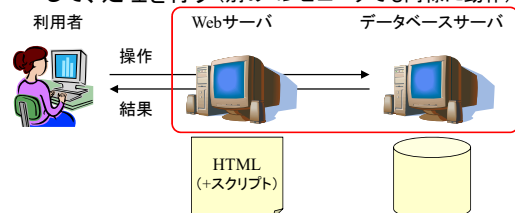
dbname には、必ず自分の **アカウント名** を入れること！

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
- ウェブサーバにファイルを置く
 - ホームディレクトリの下での public_html
 - クライアント端末からファイルを転送
- 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>

ウェブサーバ

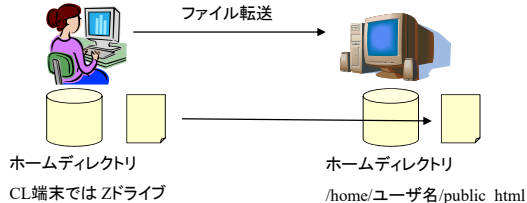
- 本演習では、データベースサーバとウェブサーバは同一のコンピュータ
 - 同一コンピュータ上にある、サーバ同士が通信して、処理を行う (別のコンピュータでも同様に動作)



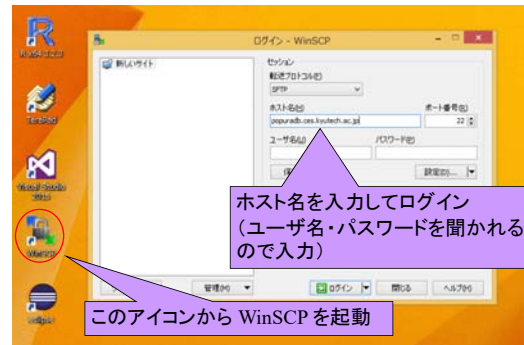
ファイルのアップロード

- ウェブサーバのホームディレクトリにファイルをアップロード

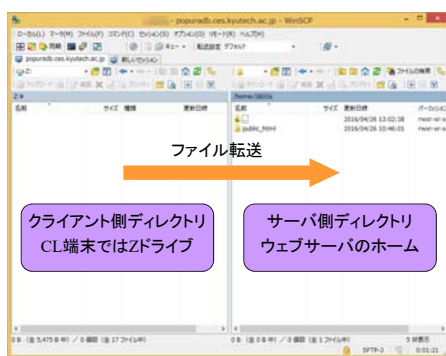
– SCP接続によるウェブサーバへのファイル転送
クライアント端末



WinSCPによるファイル転送



WinSCPによるファイル転送



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
 - 従業員データの検索

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 (employee_add.html)
 - 追加処理 (employee_add.php)
 - 追加フォーム(動的生成版) (employee_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_update_form2.php)
 - 更新処理 (employee_update.php)
 - 検索フォーム(動的生成) (employee_search_form.php)
 - 削除処理 (employee_search.php)

メニュー

- メニュー (menu.html)
 - <HTML> <HEAD> <TITLE> <BYDY>
 - ~ によるリスト
 - 各機能のページへのリンク
 ~

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 (employee_add.html)
 - 追加処理 (employee_add.php)
 - 追加フォーム(動的生成版) (employee_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_update_form2.php)
 - 更新処理 (employee_update.php)
 - 検索フォーム(動的生成) (employee_search_form.php)
 - 削除処理 (employee_search.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のスタートダッシュ -->
<?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;
}
    
```

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
$result = pg_query($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);
```

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

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

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

検索結果を表示します。

| 従業員番号 | 部門 | 氏名 | 年齢 |
|-------|----|-------|----|
| 0001 | 開発 | 渡田 信典 | 48 |
| 0002 | 営業 | 豊臣 秀吉 | 45 |
| 0003 | 総務 | 徳川 家康 | 39 |
| 0004 | 開発 | 伊田 晴孝 | 60 |
| 0005 | 営業 | 伊達 政宗 | 15 |
| 0006 | 総務 | 上杉 景勝 | 26 |
| 0007 | 開発 | 島津 重久 | 35 |

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i列のデータを取得
        $data = pg_fetch_result($result, $j, $i);

        // テーブルのj行i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");
```

テーブルの各行ごとに繰り返し

各行全体を <TR> タグで囲む

各列ごとに繰り返し

SQLの実行結果からj行i列の属性値を取得するPHPの関数

各セルを <TD> タグで囲む

変数 \$data の値を表示

一覧表示(8)

- 一覧表示 (employee_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);

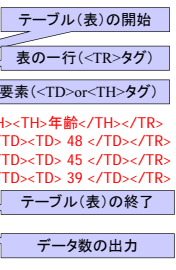
// データベースへの接続を解除
pg_close($conn);
```

文字列の中に、変数 \$rows の値が埋め込まれて出力される

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>織田 信長</TD><TD>48</TD></TR>
<TR><TD>0002</TD><TD>営業</TD><TD>豊臣 秀吉</TD><TD>45</TD></TR>
<TR><TD>0003</TD><TD>総務</TD><TD>徳川 家康</TD><TD>39</TD></TR>
.....
</TABLE><BR>
以上、7件のデータを表示しました。<BR>
<!-- ここまでで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>
    
```



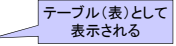
表示結果の例

- 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 | 織田 信長 | 48 |
| 0002 | 営業 | 豊臣 秀吉 | 45 |
| 0003 | 総務 | 徳川 家康 | 39 |
| 0004 | 開発 | 伊田 勝家 | 60 |
| 0005 | 営業 | 伊達 政宗 | 15 |
| 0006 | 総務 | 上杉 景勝 | 26 |
| 0007 | 開発 | 島津 家久 | 35 |

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

- 前回までの演習は完了しているものとする
- 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(＋一部変更)して、動作確認をする
- 一覧表示を行なうPHPプログラムを一部変更して、従業員の一覧が部門ごとに表示されるようにする(exmployee_list.phpを変更)

演習課題の提出

- 演習課題のテキストファイルに回答を記述して、Moodleから提出
- 提出締め切り 7月26日(木) 18:00(厳守)
 – なるべく次回の講義の前までに終わらせる

まとめ

- 前回の復習
- 前回の演習の復習
- WWWの仕組み
- HTML + PHP の基礎
- PHPによるインターフェース開発(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開発(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