

Oracle® Database

2日でPHP開発者ガイド

11g リリース2 (11.2)

部品番号: B56267-01

2009年11月

Oracle Database 2日でPHP開発者ガイド, 11g リリース2 (11.2)

部品番号: B56267-01

Oracle Database 2 Day + PHP Developer's Guide, 11g Release 2 (11.2)

原本部品番号: E10811-01

原本著者: Simon Watt

原本協力者: Christopher Jones、Simon Law、Glenn Stokol

Copyright © 2009, Oracle and/or its affiliates. All rights reserved.

制限付権利の説明

このソフトウェアおよび関連ドキュメントの使用と開示は、ライセンス契約の制約条件に従うものとし、知的財産に関する法律により保護されています。ライセンス契約で明示的に許諾されている場合もしくは法律によって認められている場合を除き、形式、手段に関係なく、いかなる部分も使用、複写、複製、翻訳、放送、修正、ライセンス供与、送信、配布、発表、実行、公開または表示することはできません。このソフトウェアのリバース・エンジニアリング、逆アセンブル、逆コンパイルは互換性のために法律によって規定されている場合を除き、禁止されています。

ここに記載された情報は予告なしに変更される場合があります。また、誤りが無いことの保証はいたしかねます。誤りを見つけた場合は、オラクル社までご連絡ください。

このソフトウェアまたは関連ドキュメントが、米国政府機関もしくは米国政府機関に代わってこのソフトウェアまたは関連ドキュメントをライセンスされた者に提供される場合は、次のNoticeが適用されます。

U. S. GOVERNMENT RIGHTS

Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

このソフトウェアは様々な情報管理アプリケーションでの一般的な使用のために開発されたものです。このソフトウェアは、危険が伴うアプリケーション（人的傷害を発生させる可能性があるアプリケーションを含む）への用途を目的として開発されていません。このソフトウェアを危険が伴うアプリケーションで使用する場合、このソフトウェアを安全に使用するために、適切な安全装置、バックアップ、冗長性（redundancy）、その他の対策を講じることは使用者の責任となります。このソフトウェアを危険が伴うアプリケーションで使用したことに起

因して損害が発生しても、オラクル社およびその関連会社は一切の責任を負いかねます。

OracleはOracle Corporationおよびその関連企業の登録商標です。 その他の名称は、それぞれの所有者の商標または登録商標です。

このソフトウェアおよびドキュメントは、第三者のコンテンツ、製品、サービスへのアクセス、あるいはそれらに関する情報を提供することがあります。 オラクル社およびその関連会社は、第三者のコンテンツ、製品、サービスに関して一切の責任を負わず、いかなる保証もいたしません。 オラクル社およびその関連会社は、第三者のコンテンツ、製品、サービスへのアクセスまたは使用によって損失、費用、あるいは損害が発生しても、一切の責任を負いかねます。

目次

[タイトルおよび著作権情報](#)

[はじめに](#)

- [対象読者](#)
- [ドキュメントのアクセシビリティについて](#)
- [関連ドキュメント](#)
- [表記規則](#)

[1 Oracle DatabaseでのPHPの導入](#)

- [目的](#)
- [サンプル・アプリケーションの概要](#)
- [リソース](#)

[2 事前準備](#)

- [必要なもの](#)
- [Oracle Databaseのインストール](#)
 - [HRユーザーのロック解除](#)
- [Apache HTTPサーバーのインストール](#)
 - [WindowsでのApacheのインストール](#)
 - [WindowsでのApacheの起動と停止](#)
 - [WindowsでのApacheインストールのテスト](#)
 - [LinuxでのApacheのインストール](#)
 - [LinuxでのApacheの起動と停止](#)
 - [LinuxでのApacheインストールのテスト](#)
- [PHPのインストール](#)
 - [WindowsでのPHPのインストール](#)
 - [LinuxでのPHPのインストール](#)
- [PHPインストールのテスト](#)

[3 接続](#)

- [「Departments」ページの構築](#)
- [データベースへの接続](#)
 - [接続で問題が発生した場合](#)
 - [他の接続方法](#)
- [データベースからの切断](#)

[4 データの問合せ](#)

- [データベース・アプリケーション・ロジックの集中化](#)
- [バインド変数を使用した問合せの作成](#)
- [データベース・レコードのナビゲート](#)
 - [ROWNUMとROW_NUMBER\(\)](#)
- [基本的な「Departments」ページの拡張](#)

[5 データの更新](#)

- [基本的な「Employees」ページの構築](#)
- [基本的な「Employees」ページの拡張](#)
- [「Departments」と「Employees」の結合](#)
- [エラー・リカバリの追加](#)
- [追加のエラー処理](#)

[6 ストアド・プロシージャおよびストアド・ファンクションの実行](#)

- [PL/SQLを使用したビジネス・ロジックの取得](#)
- [PL/SQL参照カーソルを使用した結果セットの戻し](#)

[7 イメージのロード](#)

- [BLOBを使用した従業員イメージの保存およびロード](#)
- [イメージのサイズ変更](#)

[8 グローバル・アプリケーションの構築](#)

- [OracleとPHP間の環境の確立](#)
- [文字列の操作](#)
- [ユーザーのロケールの決定](#)
- [ロケール認識の開発](#)
- [HTMLページのエンコーディング](#)
 - [HTMLページ用のページ・エンコーディングの指定](#)
 - [PHPへのページ・エンコーディングの指定](#)
- [翻訳のためのHTMLページのコンテンツの編成](#)
 - [PHPの文字列](#)
 - [静的ファイル](#)
 - [データベースのデータ](#)
- [ユーザーが考えているとおりの表記規則を使用したデータの表示](#)
 - [Oracleの日付書式](#)
 - [Oracleの数値書式](#)
 - [Oracleの言語ソート](#)
 - [Oracleのエラー・メッセージ](#)

[索引](#)

はじめに

『Oracle Database 2日でPHP開発者ガイド』では、開発者がPHPを使用してOracle Databaseにアクセスする方法について説明します。

ここでは、次の項目について説明します。

- [対象読者](#)
- [ドキュメントのアクセシビリティについて](#)
- [関連ドキュメント](#)
- [表記規則](#)

対象読者

『Oracle Database 2日でPHP開発者ガイド』では、PHPおよびOracle Databaseを使用したアプリケーション開発について説明します。

このマニュアルは、対象読者がSQL、PL/SQL、およびPHPについて大まかに理解していることを想定しています。

ドキュメントのアクセシビリティについて

オラクル社は、障害のあるお客様にもオラクル社の製品、サービスおよびサポート・ドキュメントを簡単にご利用いただけることを目標としています。オラクル社のドキュメントには、ユーザーが障害支援技術を使用して情報を利用できる機能が組み込まれています。HTML形式のドキュメントで用意されており、障害のあるお客様が簡単にアクセスできるようにマークアップされています。標準規格は改善されつつあります。オラクル社はドキュメントをすべてのお客様がご利用できるように、市場をリードする他の技術ベンダーと積極的に連携して技術的な問題に対応しています。オラクル社のアクセシビリティについての詳細情報は、Oracle Accessibility ProgramのWebサイト<http://www.oracle.com/accessibility/>を参照してください。

ドキュメント内のサンプル・コードのアクセシビリティについて

スクリーン・リーダーは、ドキュメント内のサンプル・コードを正確に読めない場合があります。コード表記規則では閉じ括弧だけを行に記述する必要があります。しかしJAWSは括弧だけの行を読まない場合があります。

外部Webサイトのドキュメントのアクセシビリティについて

このドキュメントにはオラクル社およびその関連会社が所有または管理しないWebサイトへのリンクが含まれている場合があります。オラクル社およびその関連会社は、それらのWebサイトのアクセシビリティに関しての評価や言及は行っておりません。

聴覚に障害があるお客様のOracleサポート・サービスへのアクセス

Oracleサポート・サービスに連絡するには、テレコミュニケーション・リレー・サービス (TRS) を使用してOracleサポート (+1-800-223-1711) までお電話ください。Oracleサポート・サービスの技術者が、Oracleサービス・リクエストのプロセスに従って、技術的な問題を処理し、お客様へのサポートを提供します。TRSの詳細は、<http://www.fcc.gov/cgb/consumerfacts/trs.html>を参照してください。電話番号の一覧は、<http://www.fcc.gov/cgb/dro/trsphonebk.html>を参照してください。

関連ドキュメント

詳細は、次のOracleドキュメントを参照してください。

- 『Oracle Database 2日で開発者ガイド』
- 『Oracle Database SQL言語リファレンス』

- 『Oracle Database PL/SQL言語リファレンス』
- 『SQL*Plusユーザズ・ガイドおよびリファレンス』
- 『Oracle Databaseグローバルゼーション・サポート・ガイド』

表記規則

このマニュアルでは次の表記規則を使用します。

規則	意味
太字	太字は、操作に関連するGraphical User Interface要素、または本文中で定義されている用語を示します。
イタリック体	イタリックは、ドキュメントのタイトル、強調またはユーザーが特定の値を指定するプレースホルダ変数を示します。
固定幅フォント	固定幅フォントは、段落内のコマンド、URL、サンプル内のコード、画面に表示されるテキスト、または入力するテキストを示します。

サポートおよびサービス

次の各項に、各サービスに接続するためのURLを記載します。

Oracleサポート・サービス

オラクル製品サポートの購入方法、およびOracleサポート・サービスへの連絡方法の詳細は、次のURLを参照してください。

<http://www.oracle.com/lang/jp/support/index.html>

製品マニュアル

製品のマニュアルは、次のURLにあります。

<http://www.oracle.com/technology/global/jp/documentation/index.html>

研修およびトレーニング

研修に関する情報とスケジュールは、次のURLで入手できます。

http://education.oracle.com/pls/web_prod-plq-dad/db_pages.getpage?page_id=3

その他の情報

オラクル製品やサービスに関するその他の情報については、次のURLから参照してください。

<http://www.oracle.com/lang/jp/index.html>

<http://www.oracle.com/technology/global/jp/index.html>

注意:

ドキュメント内に記載されているURLや参照ドキュメントには、Oracle Corporationが提供する英語の情報も含まれています。日本語版の情報については、前述のURLを参照してください。

1 Oracle DatabaseでのPHPの導入

PHPは、HTMLに埋め込むことができることから、特にWebの開発で有効な人気の高いスクリプト言語です。

この章の内容は次のとおりです。

- [目的](#)
- [サンプル・アプリケーションの概要](#)
- [リソース](#)

目的

このマニュアルは、PHPを使用してOracle Databaseに接続する方法およびPHPを使用してデータにアクセスし、変更を行う方法を示すチュートリアルです。

サンプル・アプリケーションの概要

このマニュアルでは、AnyCo Corpという架空の会社で人事管理（HR）サンプル・アプリケーションを開発する手順について説明します。ここで紹介するPHP言語では、フレームワークまたは抽象化レイヤーは使用していません。ただし、PHPフレームワークの普及が進んでいるため、大規模なアプリケーションを構築する場合は検討してください。

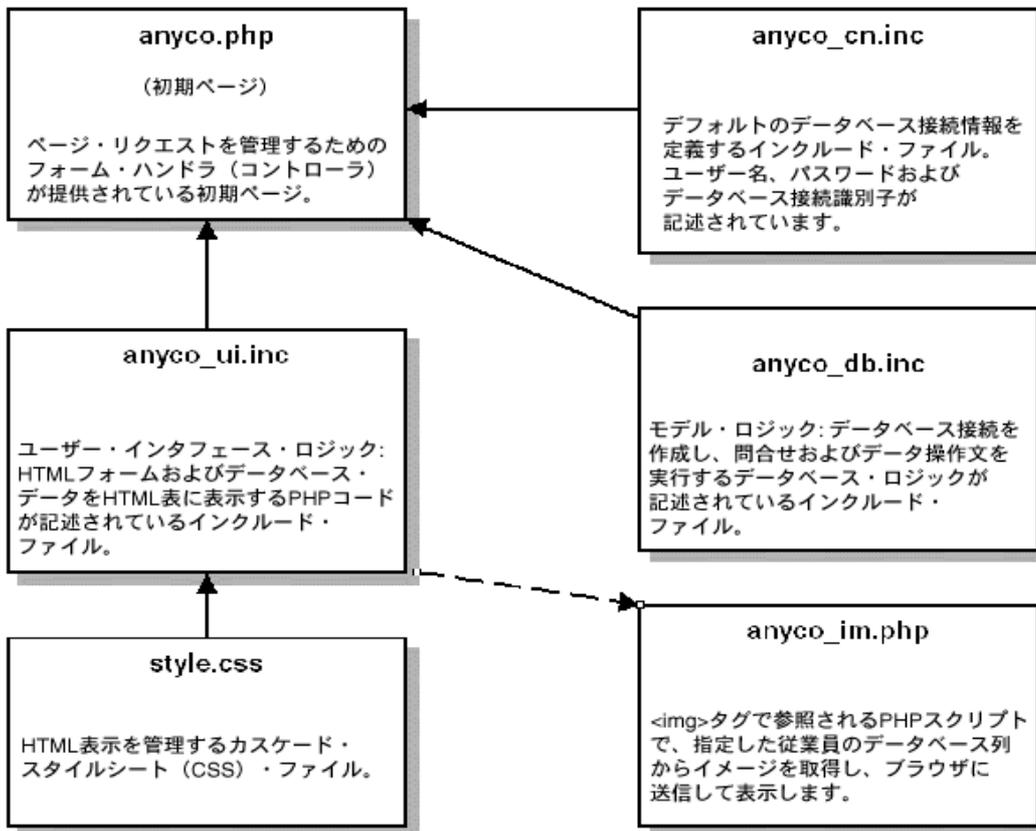
このアプリケーションでは、Oracle Databaseで用意されているHRスキーマのDEPARTMENTS表およびEMPLOYEES表に格納されている部門データおよび従業員データを管理します。このスキーマの詳細は、『Oracle Databaseサンプル・スキーマ』を参照してください。

完全なサンプル・アプリケーションでは、次のことを行います。

- PHP OCI8拡張モジュールを使用して、データベースへの接続を確立します。
- 部門データおよび従業員データをデータベースに問い合わせます。
- データの表示およびデータ内のナビゲートを行います。
- 従業員レコードを挿入、更新および削除する方法を示します。
- データ例外を処理します。
- 従業員写真をアップロードおよび表示します。

図1-1に、このアプリケーション用に開発したファイル間の関係を示します。

図1-1 人事管理サンプル・アプリケーションのコンポーネント



「図1-1 人事管理サンプル・アプリケーションのコンポーネント」の説明

サンプル・アプリケーションのファイルは次のとおりです。

- anyco.php:** このファイルには、AnyCoアプリケーションのメイン・ロジックが含まれています。表示するページを判断する制御ロジックも含まれています。このファイルによって、ナビゲーション用のセッション・データが管理されます。また、anyco_cn.inc、anyco_db.incおよびanyco_ui.incの各インクルード・ファイルの関数がコールされます。
- anyco_ui.inc:** このファイルには、HTMLページでデータおよびフォームを提示するために使用される関数が含まれています。
- anyco_cn.inc:** このファイルには、データベース接続情報、データベース・ユーザー名、パスワードおよびデータベース接続識別子の定義が含まれています。
- anyco_db.inc:** このファイルには、データベース接続を作成し、問合せを実行し、データ操作文を実行するデータベース・ロジックが含まれています。
- anyco_im.php:** このファイルには、JPEGイメージとして表示するために、データベース列からイメージを取り出してWebブラウザに送信するロジックが含まれています。
- style.css:** このファイルには、アプリケーションで生成される様々なHTMLタグのカスケード・スタイルシート（CSS）定義が含まれています。このファイルによって、アプリケーションのルック・アンド・フィールが管理されます。

接尾辞が.incのファイルは、他のPHPファイルにインクルードされるPHPコード・ファイルです。

接尾辞が.phpのファイルは、Webブラウザにロードできます。

PHPアプリケーション・ソース・ファイルは、PHP開発をサポートするツール（テキスト・エディタなど）で作成および編集できます。

各章で示すコードは、その前章で作成したファイルに基づいています。

リソース

次のOTN (Oracle Technology Network) のWebサイトでは、ユーザーにとって有効な追加情報が提供されています。

- PHP Developer Center

<http://www.oracle.com/technology/tech/php/index.html>

- Oracle Databaseドキュメント・ライブラリ

<http://www.oracle.com/technology/documentation>

- Oracle SQL Developer Center

http://www.oracle.com/technology/products/database/sql_developer/

2 事前準備

この章では、Oracle DatabaseおよびPHP環境をインストールし、テストする方法について説明します。内容は次のとおりです。

- [必要なもの](#)
- [Oracle Databaseのインストール](#)
- [Apache HTTPサーバーのインストール](#)
- [PHPのインストール](#)
- [PHPインストールのテスト](#)

必要なもの

Oracle DatabaseおよびPHP環境をインストールするには、次のものがが必要です。

- Oracle Databaseサーバー（このチュートリアル用のサンプル・データを含む）
- Oracle Databaseクライアント・ライブラリ（Oracle Databaseソフトウェアに含まれている。また、Oracle Instant Clientを使用してインストール可能）
- Apache HTTPサーバー（通常は、Linuxコンピュータにすでにインストール済）
- PHP（PHP Hypertext Preprocessor）
- PHPコードを編集するためのテキスト・エディタ。オプションのPHP拡張モジュールに対応しているコード・エディタ（Oracle JDeveloperなど）も使用できます。

Oracle Databaseのインストール

Oracle Databaseサーバーをコンピュータにインストールする必要があります。このチュートリアルで使用するサンプル・データは、デフォルトでインストールされます。これがサンプル・スキーマのHRコンポーネントとなります。

このチュートリアルでは、Oracle SQL Developerがデータベース・タスクの実行に使用するグラフィカル・ユーザー・インタフェースとなります。Oracle SQL Developerは、データベース開発用の無料のグラフィカルなツールです。

参照:

- HRサンプル・スキーマの詳細は、『Oracle Databaseサンプル・スキーマ』を参照してください。
- Oracle SQL DeveloperのWebページ

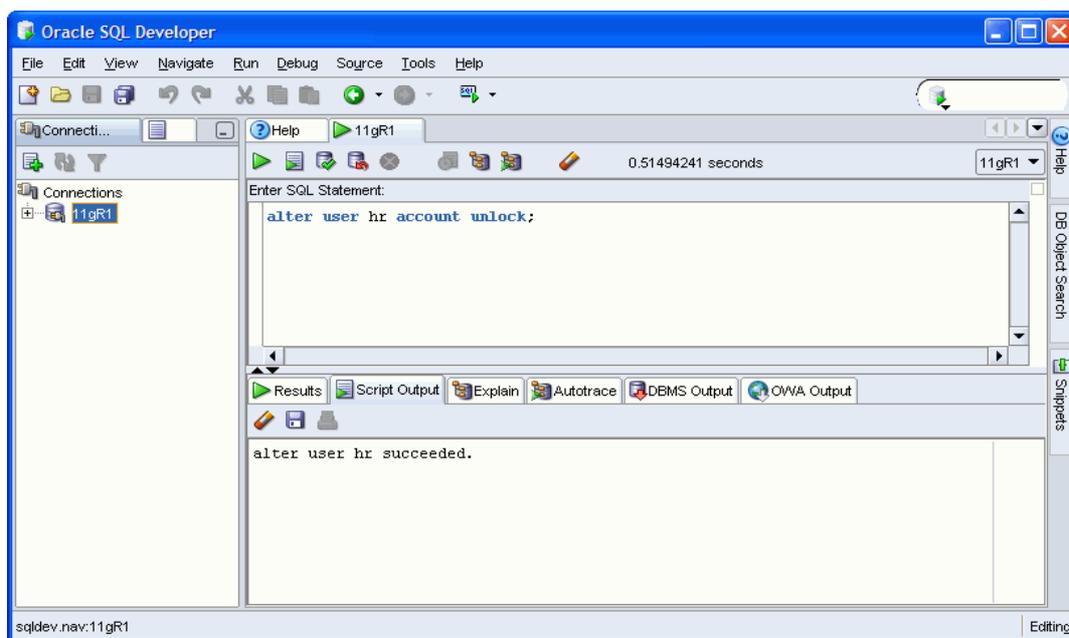
http://www.oracle.com/technology/products/database/sql_developer/

HRユーザーのロック解除

PHPアプリケーションは、HRユーザーとしてデータベースに接続します。DBA権限を持つユーザーとして、HRアカウントをロック解除する必要があります。HRユーザーをロック解除するには、次の手順を実行します。

1. SQL Developerを起動し、Oracle Databaseへの接続をオープンします。
2. **system**ユーザーとしてOracleデータベースにログインします。
3. SQLワークシートまたはSQL*Plusを開き、次のSQL文を実行します。

```
alter user hr account unlock;
```



[chap2_unlock.gifの説明](#)

Oracle Databaseアカウントのロック解除の詳細は、『Oracle Database 2日でデータベース管理者』の第6章「ユーザーおよびセキュリティの管理」を参照してください。

参照:

- Oracle Databaseドキュメント

<http://www.oracle.com/technology/documentation>

Apache HTTPサーバーのインストール

PHPをインストールする前に、Apacheをインストールする必要があります。通常、ApacheはLinuxコンピュータにデフォルトでインストールされています。Apacheソフトウェアをダウンロードする前に、[「LinuxでのApacheインストールのテスト」](#)を参照してください。

WindowsまたはLinux用のApache HTTPサーバーを入手するには、次の手順を実行します。

1. Webブラウザに次のURLを入力します。

<http://httpd.apache.org/download.cgi>

2. Windowsの場合は、**apache_2.2.12-win32-x86-no_ssl.msi**をクリックします。Linuxの場合は、**httpd-2.2.12.tar.bz2**をクリックします。
3. ダウンロードしたファイルを一時ディレクトリ（Windowsの場合は`c:\¥tmp`、Linuxの場合は`¥tmp`）に保存します。

WindowsでのApacheのインストール

この項では、WindowsにApache HTTPサーバーをインストールする方法について説明します。

ファイル名および抽出ディレクトリは、現行バージョンに基づいて決定されます。この手順では、インストールしているバージョンのディレクトリ名を使用するようにしてください。

Apacheをインストールするには、管理者ユーザーである必要があります。

Apacheをインストールするには、ファイルをダブルクリックし、ウィザードに従います。

デフォルトのインストール・ディレクトリは、一般的に`C:\¥Program Files¥Apache Group`です。チュートリアルの際は、このディレクトリを前提としています。別のディレクトリにインストールする場合は、ディレクトリに注意して、チュートリアルで示されるパスを読み替える必要があります。

WindowsでのApacheの起動と停止

「スタート」メニュー・オプションを使用して、Apacheを起動できます。コンソール・ウィンドウが開き、エラー・メッセージがあれば表示されます。エラー・メッセージは、`C:\¥Program Files¥Apache Group¥Apache2¥logs¥error.log`にも書き込まれる場合があります。

ApacheMonitorユーティリティを使用して、Apacheを起動することもできます。ApacheMonitorユーティリティはシステム・トレイにアイコンで表示されます。または、Apache binディレクトリに移動してApacheMonitor.exeをダブルクリックして使用します。デフォルトのインストールでは、Apacheのbinディレクトリは`c:\¥Program Files¥Apache Group¥Apache2¥bin`です。

Windowsのサービスを使用して、Apacheを起動することもできます。Windowsのサービスにアクセスするには、Windowsの「スタート」メニューで、「スタート」→「コントロールパネル」→「管理ツール」→「サービス」を選択します。「標準」タブを選択します。「Apache2 HTTP Server」を右クリックし、「再起動」を選択します。

エラーが発生した場合は、`httpd.conf`および`php.ini`ファイルを再確認します。

WindowsでのApacheインストールのテスト

Apache HTTPサーバーのインストールをテストするには、次の手順を実行します。

1. ApacheをインストールしたコンピュータでWebブラウザを起動します。
2. 次のURLを入力します。

```
http://localhost/
```

Webブラウザに、次のようなページが表示されます。

If you can see this, it means that the installation of the [Apache web server](#) software on this system was successful. You may now add content to this directory and replace this page.

Seeing this instead of the website you expected?

This page is here because the site administrator has changed the configuration of this web server. Please **contact the person responsible for maintaining this server with questions**. The Apache Software Foundation, which wrote the web server software this site administrator is using, has nothing to do with maintaining this site and cannot help resolve configuration issues.

The Apache [documentation](#) has been included with this distribution.

You are free to use the image below on an Apache-powered web server. Thanks for using Apache!



[chap2_test_install_013.gifの説明](#)

このページが表示されない場合は、Apache構成を確認します。一般的な問題としては、Apacheが実行されていない、Apacheがデフォルト以外のポートをリスニングしているなどの問題が考えられます。

LinuxでのApacheのインストール

この項では、LinuxにApache HTTPサーバーをインストールする方法について説明します。

ファイル名および抽出ディレクトリは、現行バージョンに基づいて決定されます。この手順では、インストールしているバージョンのディレクトリ名を使用するようにしてください。

通常、ApacheはLinuxにすでにインストールされています。[「LinuxでのApacheインストールのテスト」](#)の後にApacheがインストールされていないことがわかった場合、次の手順を実行してApache HTTPサーバーをインストールします。

1. httpd-2.2.12.tar.bz2ファイルをダウンロードしたディレクトリに移動します。
2. rootユーザーとしてログインし、次のコマンドを実行します。

```
# tar -jxvf httpd-2.2.12.tar.bz2
# cd httpd-2.2.12
# export ORACLE_HOME=/usr/lib/oracle/app/oracle/product/10.2.0/server
# ./configure ¥
    --prefix=/usr/local/apache ¥
    --enable-module=so ¥
# make
# make install
```

--enable-module=soオプションは、PHPをDynamic Shared Object (DSO) としてコンパイルすることを許可します。--prefix=オプションは、make installコマンドで使用されるApacheインストール・ディレクトリを設定します。

LinuxでのApacheの起動と停止

apachectlスクリプトを使用して、Apacheを起動および停止します。

Apacheを起動するには、次のapachectlスクリプトを実行します。

```
# /usr/local/apache/bin/apachectl start
```

Apacheを停止するには、次のapachectlスクリプトを実行します。

```
# /usr/local/apache/bin/apachectl stop
```

Apacheを起動する場合、ORACLE_HOMEが定義されている必要があります。他の必須のOracle環境変数も、Apacheの起動前に設定されている必要があります。これらは、\$ORACLE_HOME/bin/oracle_env.shスクリプトまたは/usr/local/bin/oraenvスクリプトで設定される変数と同じです。

次のように、Apacheを起動する便利なスクリプトを作成できます。

```
#!/bin/sh
ORACLE_HOME=/usr/lib/oracle/app/oracle/product/10.2.0/server
export ORACLE_HOME
echo "Oracle Home: $ORACLE_HOME"
echo Starting Apache
/usr/local/apache/bin/apachectl start
```

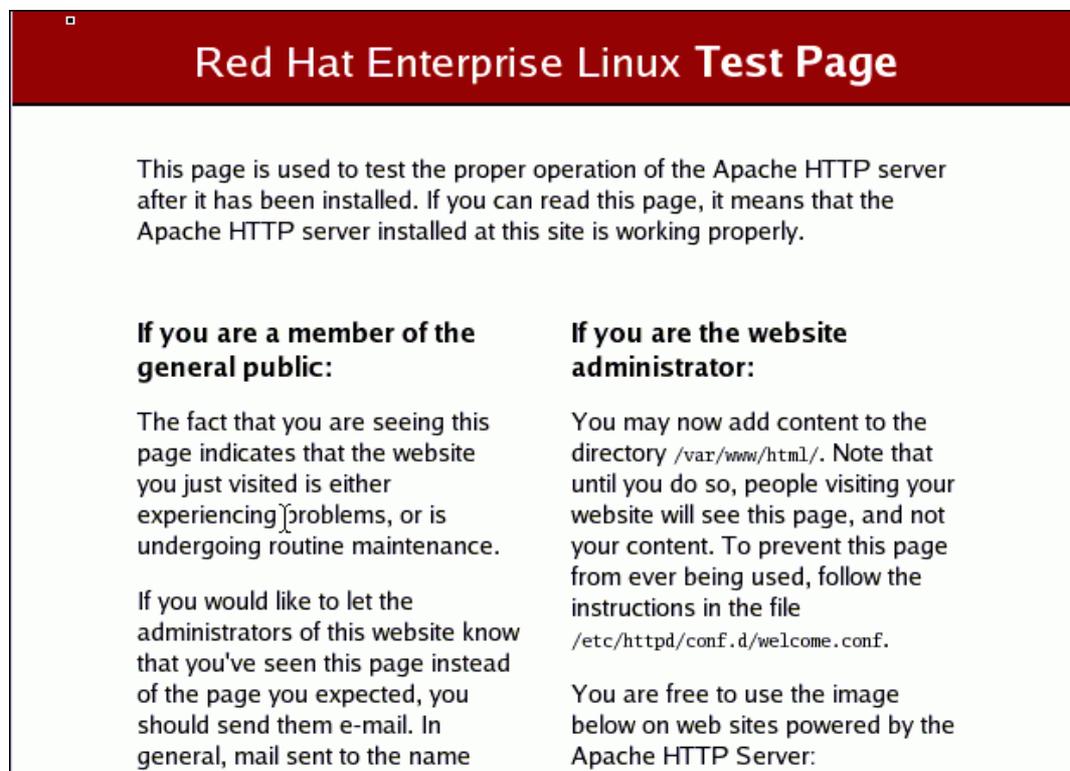
LinuxでのApacheインストールのテスト

Apache HTTPサーバーのインストールをテストするには、次の手順を実行します。

1. ApacheをインストールしたホストでWebブラウザを起動し、次のURLを入力します。

```
http://localhost/
```

Webブラウザに、次のようなページが表示されます。



このページが表示されない場合は、Apache構成を確認します。一般的な問題としては、Apacheが実行されていない、Apacheがデフォルト以外のポートをリスニングしているなどの問題が考えられます。

2. PHPファイルにアクセスできるようにするために、デフォルトのApache HTTPサーバー構成ファイルにパブリック仮想ディレクトリを `public_html` と設定します。 任意のエディタを使用してApache構成ファイル `/etc/httpd/conf/httpd.conf` (Linuxにインストールした場合は、ディレクトリが異なることがあります) を開き、次の行の先頭にあるシャープ記号 (#) を削除します。

```
UserDir public_html
```

この例では、Apache `httpd.conf` ファイルに次の行が含まれています。

```
<IfModule mod_userdir.c>
#
# UserDir is disabled by default since it can confirm the presence
# of a username on the system (depending on home directory
# permissions).
#
#UserDir disable

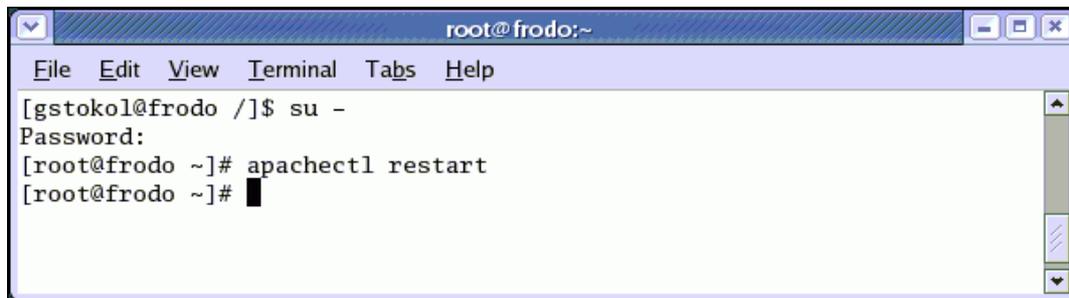
#
# To enable requests to /~user/ to serve the user's public_html
# directory, remove the "UserDir disable" line above, and uncomment
# the following line instead:
#
UserDir public_html
</IfModule>
```

これによって、Webブラウザで、システムの登録ユーザーを使用してHTTPリクエストを作成し、そのユーザーの `$HOME/public_html` ディレクトリにあるファイルを処理できるようになります。 次に例を示します。

```
http://localhost/~user/
```

3. 新しいApache構成ファイルを使用するには、コマンド・ウィンドウに次のコマンドを入力してApacheを再起動します。

```
su -
Password: <enter your su (root) password>
apachectl restart
```



[chap2_install_002.gifの説明](#)

Apache HTTPサーバーが起動しない場合は、エラー・ログ・ファイルを調べて原因を特定します。 構成エラーである可能性があります。

4. コマンド・ウィンドウに次のコマンドを入力して、(root以外で) ログインし、`$HOME`ディレクトリに `public_html` サブディレクトリを作成します。

```
mkdir $HOME/public_html
```



```
gstokol@frodo:~  
File Edit View Terminal Tabs Help  
[gstokol@frodo /]$ su - gstokol  
Password:  
[gstokol@frodo ~]$ mkdir $HOME/public_html  
[gstokol@frodo ~]$ exit
```

[chap2_install_003.gifの説明](#)

PHPのインストール

WindowsまたはLinux用のPHPを入手するには、次の手順を実行します。

1. Webブラウザに次のURLを入力します。

<http://www.php.net/downloads.php>

2. Windowsの場合は、「Windows Binaries」の下の「**5.2.10 zip package**」をクリックします。Linuxの場合は、「Complete Source Code」の下の「**php-5.2.10.tar.bz2**」をクリックします。
3. ダウンロードしたファイルを一時ディレクトリ（Windowsの場合は`c:\¥tmp`、Linuxの場合は`¥tmp`）に保存します。

WindowsでのPHPのインストール

この項では、WindowsにPHPをインストールする方法について説明します。

ファイル名および抽出ディレクトリは、現行バージョンに基づいて決定されます。この手順では、インストールしているバージョンのディレクトリ名を使用するようにしてください。

PHPをインストールするには、管理者ユーザーである必要があります。PHPをインストールするには、次の手順を実行します。

1. Windowsのエクスプローラで、PHP 5.2.10 zipファイルをダウンロードしたディレクトリに移動します。
2. `C:\¥php-5.2.10`というディレクトリにPHPパッケージを解凍します。
3. `php.ini-recommended`を`C:\¥Program Files\¥Apache Group\¥Apache2\¥conf\¥php.ini`にコピーします。
4. `php.ini`を編集して、次の変更を行います。
 - `extension_dir`を`C:\¥php-5.2.10\¥ext`に変更します。このディレクトリには`php_oci8.dll`などのPHP拡張モジュールが含まれています。
 - 行頭のセミコロンを削除します。
`extension=php_oci8.dll`
 - PHPディレクティブ`display_errors`を`On`に設定します。この設定は、テスト時にコードの問題点の確認に役立ちます。
5. ファイル`httpd.conf`を編集して、次の行を追加します。必ず、バックスラッシュ (`¥`) ではなくスラッシュ (`/`) を使用してください。

#

```

# This will load the PHP module into Apache
#
LoadModule php5_module c:/php-5.2.10/php5apache2.dll

#
# This next section will call PHP for .php, .phtml, and .phps files
#
AddType application/x-httpd-php .php
AddType application/x-httpd-php .phtml
AddType application/x-httpd-php-source .phps

#
# This is the directory containing php.ini
#
PHPIniDir "C:/Program Files/Apache Group/Apache2/conf "

```

6. Apacheサーバーを再起動すると、PHPインストールのテストが可能になります。

「スタート」メニュー・オプションを使用して、Apacheを起動できます。 コンソール・ウィンドウが開き、エラー・メッセージがあれば表示されます。 エラー・メッセージは、C:\Program Files\Apache Group\Apache2\logs\error.logにも書き込まれる場合があります。

ApacheMonitorユーティリティを使用して、Apacheを起動することもできます。 ApacheMonitorユーティリティはシステム・トレイにアイコンで表示されます。 または、Apache binディレクトリに移動してApacheMonitor.exeをダブルクリックして使用します。 デフォルトのインストールでは、Apacheのbinディレクトリはc:\Program Files\Apache Group\Apache2\binです。

Windowsのサービスを使用して、Apacheを起動することもできます。 Windowsのサービスにアクセスするには、Windowsの「スタート」メニューで、「スタート」→「コントロールパネル」→「管理ツール」→「サービス」を選択します。「標準」タブを選択します。「Apache2 HTTP Server」を右クリックし、「再起動」を選択します。

エラーが発生した場合は、httpd.confおよびphp.iniファイルを再確認します。

LinuxでのPHPのインストール

この項では、LinuxにPHPをインストールする方法について説明します。

ファイル名および抽出ディレクトリは、現行バージョンに基づいて決定されます。 この手順では、インストールしているバージョンのディレクトリ名を使用するようにしてください。

PHPをインストールするには、次の手順を実行します。

1. php-5.2.10.tar.bz2ファイルをダウンロードしたディレクトリに移動します。
2. rootユーザーとしてログインし、次のコマンドを実行します。

```

# tar -jxvf php-5.2.10.tar.bz2
# cd php-5.2.10
# export ORACLE_HOME=/usr/lib/oracle/app/oracle/product/10.2.0/server
# ./configure ¥
    --with-oci8=$ORACLE_HOME ¥
    --with-apxs2=/usr/local/apache/bin/apxs ¥
    --with-config-file-path=/usr/local/apache/conf ¥
    --enable-sigchild
# make
# make install

```

ORACLE_HOMEの値を確認して、Oracleのバージョンとインストールが反映されていることを確認します。

ファイアウォールの内側にいる場合、make installを実行する前に環境変数http_proxyを使用しているプロキシ・サーバーに設定する必要があります。 この設定によって、PHPのPEARコンポーネントのインストールが可能になります。

3. PHPの提供する初期化ファイルをコピーします。

```
# cp php.ini-recommended /usr/local/apache/conf/php.ini
```

このphp.iniは、テスト時にコードの問題点を確認できるように、PHPディレクティブdisplay_errorsをOnに設定する際に役立ちます。

4. /usr/local/apache/conf/httpd.confのApacheの構成ファイルを編集して、次の行を追加します。

```
#
# This next section will call PHP for .php, .phtml, and .phps files
#
AddType application/x-httpd-php .php
AddType application/x-httpd-php .phtml
AddType application/x-httpd-php-source .phps

#
# This is the directory containing php.ini
#
PHPIniDir "/usr/local/apache/conf"
```

PHPのインストールでLoadModule行が挿入されていなかった場合は、次のように追加します。

```
LoadModule php5_module modules/libphp5.so
```

5. PHPインストールをテストするために、次のコマンドを実行してApacheサーバーを再起動します。

```
# /usr/local/apache/bin/apachectl start
```

エラーが発生した場合は、画面に表示されます。エラーは、/usr/local/apache/logs/error_logにも書き込まれる場合があります。問題があった場合は、httpd.confおよびphp.iniを再確認します。

PHPインストールのテスト

PHPのインストールをテストするには、次の手順を実行します。

1. chap2というサブディレクトリを作成します。アプリケーション・ファイル用のディレクトリを作成し、新しく作成したディレクトリに変更するには、コマンド・ウィンドウに次のコマンドを入力します。

Windowsの場合：

```
mkdir "c:\program files\Apache Group\Apache2\htdocs\chap2"
cd c:\program files\Apache Group\Apache2\htdocs\chap2
```

Linuxの場合：

```
mkdir $HOME/public_html/chap2
cd $HOME/public_html/chap2
```

別の場所にファイルを作成する場合は、作業ディレクトリ名およびURLと一致するように、ファイルの編集および実行の手順を変更する必要があります。

2. 次のHTMLテキストを含むhello.phpというファイルを作成します。

```
<?php
echo "Hello, world!";
?>
```

3. Webブラウザを開き、次のURLを入力します。

Windowsの場合:

```
http://localhost/chap2/hello.php
```

Linuxの場合:

```
http://localhost/~<username>/chap2/hello.php
```

ブラウザに「Hello, world!」という行が表示されます。

A screenshot of a web browser window with a light green background. The text "Hello, world!" is displayed in a bold, black, serif font, centered in the upper portion of the window. The window has a thin black border.

Hello, world!

[chap2_hello_001.gifの説明](#)

3 接続

この章では、Oracle Databaseへの接続および切断を行うPHP関数を実装するHRアプリケーション・ファイルを作成します。また、問合せを実行してデータベース接続が正常に確立されたことを確認できるPHP関数も開発します。

この章では、関数をコールして「Departments」ページのヘッダーおよびフッターを生成するPHPファイルを作成および変更する手順についても説明します。「Departments」ページのフッター・セクションには日時が含まれています。

この章の内容は次のとおりです。

- [「Departments」ページの構築](#)
- [データベースへの接続](#)
- [データベースからの切断](#)

注意:

ここでは、わかりやすくするために、サンプル・アプリケーション・コードにユーザー名およびパスワードが書き込まれています。デプロイするアプリケーションでは、アプリケーションのソース・コードにユーザー名およびパスワードの文字列を直接コーディングしないことをお勧めします。ユーザーにユーザー名およびパスワードの入力を求めるダイアログ・ボックスを実装する方法などのより安全な方法を使用することをお勧めします。

セキュリティ機能およびセキュリティ・プラクティスの詳細は、『Oracle Databaseセキュリティ・ガイド』および開発環境のドキュメントを参照してください。

「Departments」ページの構築

この項では、アプリケーションの最初の画面用の関数およびスタイルを作成します。

「Departments」ページを構築するには、次の手順を実行します。

1. アプリケーション・ファイル用のディレクトリを作成し、新しく作成したディレクトリに変更するには、コマンド・ウィンドウに次のコマンドを入力します。

Windowsの場合:

```
mkdir c:\program files\Apache Group\Apache2\htdocs\chap3
cd c:\program files\Apache Group\Apache2\htdocs\chap3
```

Linuxの場合:

```
mkdir $HOME/public_html/chap3
cd $HOME/public_html/chap3
```

別の場所にファイルを作成する場合は、作業ディレクトリ名およびURLと一致するように、ファイルの編集および実行の手順を変更する必要があります。

2. アプリケーション・ユーザー・インタフェースの開発を開始するには、任意のテキスト・エディタを使用してanyco_ui.incというファイルを作成し、ui_print_header()とui_print_footer()の2つの関数をそれぞれのパラメータとともにそのファイルに含めて、アプリケーションのWebページでヘッダー・セクションおよびフッター・セクションの一貫性を保持できるようにします。

```

<?php
function ui_print_header($title)
{
    $title = htmlentities($title);
    echo <<<END
    <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
        "http://www.w3.org/TR/html4/strict.dtd">
    <html>
    <head>
        <meta http-equiv="Content-Type"
            content="text/html; charset=ISO-8859-1">
        <link rel="stylesheet" type="text/css" href="style.css">
        <title>Any Co.: $title</title>
    </head>
    <body>
    <h1>$title</h1>
    END;
}

function ui_print_footer($date)
{
    $date = htmlentities($date);
    echo <<<END
    <div class="footer">
        <div class="date">$date</div>
        <div class="company">Any Co.</div>
    </div>
    END;
}
?>

```

- このアプリケーション設計では、PHP関数定義を使用して、再利用可能なコードをモジュール化します。
- `anycoui.inc`の各関数では、ヒア・ドキュメントというPHPの言語要素を使用します。これによって、次の2行間に任意のサイズのHTML形式のテキストを配置できます。

```

echo <<<END
END;

```

- `END;`行の先頭には空白を配置しないでください。配置すると、残りのドキュメントが、出力対象テキストの一部として処理されます。
- ヒア・ドキュメントの本体内に記述されているすべてのPHPパラメータ（`$title`や`$date`など）は、それぞれの値に置き換えられます。
- PHP関数`htmlentities()`は、ユーザー指定のテキストに誤ってHTMLマークアップが含まれ、出力形式に影響を与えるのを防ぐために使用します。

3. PHPファイルでは、ブラウザでのHTMLの表示スタイルを指定するために`style.css`というカスケード・スタイルシート（CSS）・ファイルが使用されます。

chap3ディレクトリに、次のCSSテキストを含む`style.css`ファイルを作成します。

```

body
{ background: #CCCCFF;
  color: #000000;
}

h1
{ border-bottom: solid #334B66 4px;
  font-size: 160%;}

table
{ padding: 5px;}

td
{ border: solid #000000 1px;
  text-align: left;
  padding: 5px;}

```

```

th
{ text-align: left;
padding: 5px; }

.footer
{ border-top: solid #334B66 4px;
font-size: 90%; }

.company
{ padding-top: 5px;
float: right; }

.date
{ padding-top: 5px;
float: left; }

```

4. ユーザー・インタフェース関数をコールするには、次のテキストを含むanyco.phpファイルを作成します。

```

<?php
require('anyco_ui.inc');

ui_print_header('Departments');
ui_print_footer(date('Y-m-d H:i:s'));

?>

```

PHPコマンドrequire()は、anyco_ui.incをインクルードするために使用します。新しい関数をコールしてHTML出力を生成できません。

5. anyco.phpファイルをテストするには、ブラウザに次のURLを入力します。

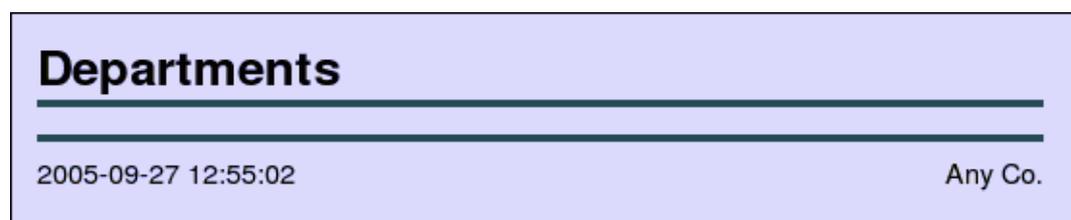
Windowsの場合:

`http://localhost/chap3/anyco.php`

Linuxの場合:

`http://localhost/~<username>/chap3/anyco.php`

生成されるWebページは、次のようになります。



[chap3_test_install_005.gifの説明](#)

日時がページのフッター・セクションに表示されます。

データベースへの接続

この項では、部門データを表示できるように「Departments」画面にデータベース接続を追加します。

アプリケーションにデータベース接続を追加するには、次の手順を実行します。

データベース接続を確立するには、次の3つの文字列パラメータを指定したoci_connect()関数を使用します。

```
$conn = oci_connect($username, $password, $db)
```

1つ目および2つ目のパラメータはそれぞれデータベース・ユーザー名およびパスワードです。3つ目のパラメータはデータベース接続識別子です。oci_connect()関数は、他のOCI8コールに必要な接続リソースを戻します。ただし、エラーが発生した場合はFALSEを戻します。戻された接続識別子は、\$connという変数に格納されます。

1. 次のパラメータ値が指定されているデータベース接続を追加するためにanyco.phpファイルを編集します。
 - ユーザー名はhrです。
 - この例のパスワードはhrです。実際には、HRユーザーの実際のパスワードを使用してください。
 - Oracle接続識別子は//localhost/orclです。
2. oci_connect()コールが使用可能なデータベース接続を戻すことを検証するためにanyco.phpファイルを編集します。oci_connect()へのコールで取得したデータベース接続識別子およびDEPARTMENTS表のすべての行を選択する問合せ文字列の2つのパラメータをとるdo_query()関数を記述します。
3. 問合せを実行するための準備を整えるためにanyco.phpファイルを編集します。oci_parse()コールを追加します。oci_parse()関数には、接続識別子および問合せ文字列の2つのパラメータが含まれています。この関数は、問合せを実行し、生成されたデータ行をフェッチするために必要な文識別子を戻します。エラーが発生した場合は、FALSEを戻します。
4. 問合せを実行するためにanyco.phpファイルを編集します。oci_execute()関数へのコールを追加します。oci_execute()関数は、1つ目のパラメータに指定された文識別子に関連付けられている文を実行します。2つ目のパラメータには、実行モードを指定します。OCI_DEFAULTは、文を自動的にコミットしないことを示すために使用します。デフォルトの実行モードはOCI_COMMIT_ON_SUCCESSです。oci_execute()関数は、正常に完了した場合はTRUEを戻し、そうでない場合はFALSEを戻します。
5. 実行した問合せに対して生成されるすべての行をフェッチするためにanyco.phpファイルを編集します。whileループおよびoci_fetch_array()関数へのコールを追加します。oci_fetch_array()関数は、結果データの次の行を戻します。次の行が存在しない場合は、FALSEを戻します。oci_fetch_array()関数の2つ目のパラメータOCI_RETURN_NULLSは、NULLデータベース・フィールドがPHP NULL値として戻されることを示しています。

データの各行は数値配列として戻されます。コードでは、配列のループにPHPのforeach構文が使用され、各列値が表の行要素内のHTML表セルに出力されます。列値がNULLの場合は、改行なしスペースが出力されます。それ以外の場合は、列値が出力されます。

手順1~5までの編集を終えると、anyco.phpファイルは次のようになります。

```
<?php // File: anyco.php

require('anyco_ui.inc');

// Create a database connection
$conn = oci_connect('hr', 'hr', '//localhost/orcl');

ui_print_header('Departments');
do_query($conn, 'SELECT * FROM DEPARTMENTS');
ui_print_footer(date('Y-m-d H:i:s'));

// Execute query and display results
function do_query($conn, $query)
{
    $stid = oci_parse($conn, $query);
    $r = oci_execute($stid, OCI_DEFAULT);

    print '<table border="1">';
    while ($row = oci_fetch_array($stid, OCI_ASSOC+OCI_RETURN_NULLS)) {
        print '<tr>';
        foreach ($row as $item) {
            print '<td>'.
                ($item !== null ? htmlentities($item) : '&nbsp;');</td>';
        }
        print '</tr>';
    }
    print '</table>';
}
```

```
}
```

```
?>
```

6. `anyco.php`に行った変更をテストするために、変更後の`anyco.php`ファイルを保存します。ブラウザのウィンドウに、次のURLを入力します。

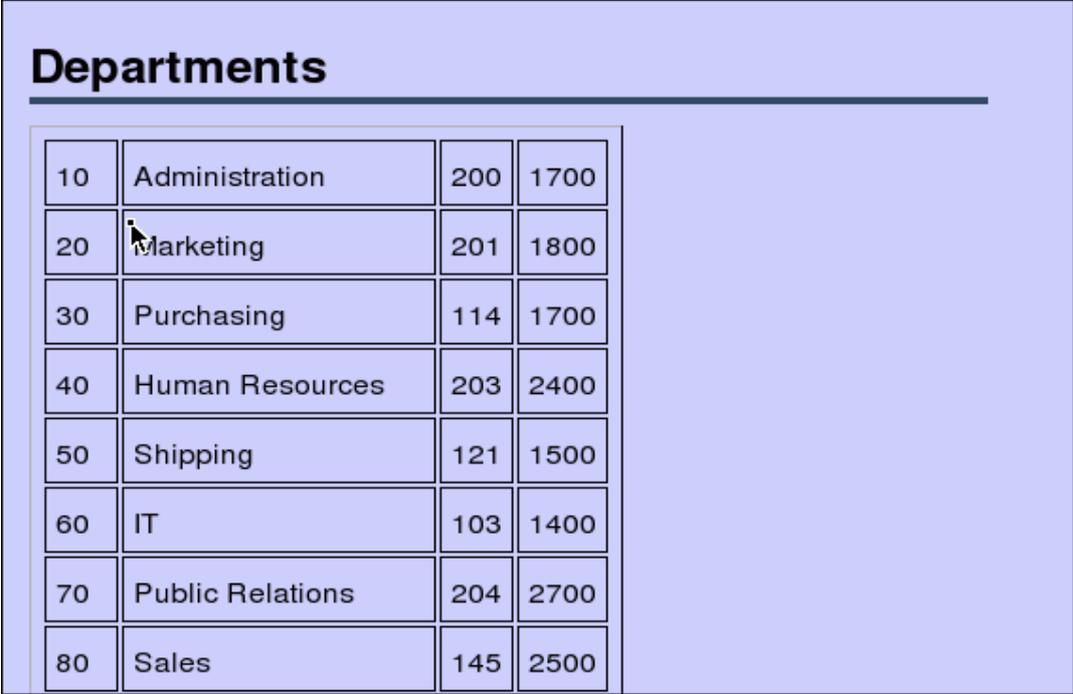
Windowsの場合:

```
http://localhost/chap3/anyco.php
```

Linuxの場合:

```
http://localhost/~<username>/chap3/anyco.php
```

ブラウザ・ウィンドウに、次のようなページが戻されます。



Departments			
10	Administration	200	1700
20	Marketing	201	1800
30	Purchasing	114	1700
40	Human Resources	203	2400
50	Shipping	121	1500
60	IT	103	1400
70	Public Relations	204	2700
80	Sales	145	2500

[chap3_db_connect_001.gifの説明](#)

EMPLOYEESのデータを問い合わせる場合は、`do_query()`関数コールの問合せを次のように変更することもできます。

```
do_query($conn, 'SELECT * FROM EMPLOYEES');
```

接続で問題が発生した場合

ユーザー名、パスワードおよび接続文字列が有効であることを確認します。接続文字列`//localhost/orcl`で、Oracle簡易接続構文を使用します。Oracle Net `tnsnames.ora`ファイルを使用して接続先のデータベースを指定する場合は、`oci_connect()`関数の3つ目のパラメータとしてネットワーク別名を使用します。

エラーが表示されていない場合は、PHPディレクティブ`display_errors`をONに設定し、`error_reporting`ディレクティブを`E_ALL|E_STRICT`に設定します。

PHPコードに問題があり、デバッガを使用していない場合は、PHPの`var_dump()`関数を使用して変数を調べることができます。次に例を示します。

```
print '<pre>';
```

```
var_dump($r);
print '</pre>';
```

他の接続方法

一部のアプリケーションでは、永続接続を使用すると、スクリプトがコールされるたびに再接続する必要がなくなるため、パフォーマンスが向上します。この場合、Apacheの構成によっては、多くのデータベース接続が同時にオープンしたままになることがあります。接続のパフォーマンス上のメリットとデータベース・サーバーで発生するオーバーヘッドとのバランスをとる必要があります。

永続接続は、OCI8のoci_pconnect()関数によって確立されます。PHP初期化ファイルの設定を使用すると、永続接続の存続時間を制御できます。この設定には、次のものが含まれています。

oci8.max_persistent: プロセスごとの永続接続の数を制御します。

oci8.persistent_timeout: プロセスでアイドル状態の永続接続が保持される時間（秒単位）を指定します。

oci8.ping_interval: 永続接続をpingしてその妥当性を確認する前に待機する必要がある時間（秒単位）を指定します。

詳細は、次のPHPのリファレンス・マニュアルを参照してください。

<http://www.php.net/manual/en/ref.oci8.php>

接続プーリングの詳細は、『Oracle Call Interfaceプログラマーズ・ガイド』および『Oracle Database Net Services管理者ガイド』のOCIでの接続プーリングに関する項を参照してください。

データベースからの切断

PHPエンジンは、永続接続が確立されていないかぎり、スクリプトの終了時に自動的にデータベース接続をクローズします。明示的に非永続データベース接続をクローズする場合は、oci_connect()コールによって戻される接続識別子でoci_close() OCI関数をコールできます。次に例を示します。

```
<?php
$conn = oci_connect('hr', '<your_password>', '//localhost/orcl');
...
oci_close($conn);
...
?>
```

PHPは変数の追跡に参照カウント・メカニズムを使用するため、接続を参照するすべてのPHP変数の設定が解除されるか、変数のスコープ外に出るまで、実際にはデータベース接続が閉じられない場合があります。

4 データの問合せ

この章では、「Departments」ページに情報を追加して、第3章で作成したAnycos HRアプリケーションを拡張します。また、特定の部門の従業員レコードに対して問合せ、挿入、更新および削除を行う機能も実装します。

この章の内容は次のとおりです。

- [データベース・アプリケーション・ロジックの集中化](#)
- [バインド変数を使用した問合せの作成](#)
- [データベース・レコードのナビゲート](#)
- [基本的な「Departments」ページの拡張](#)

データベース・アプリケーション・ロジックの集中化

この項では、PHPアプリケーションにインクルードできるようにデータベース・アクセス・ロジックを個別のファイルに移動することによって、アプリケーション・コードを変更します。

1. 第3章で作成したファイルをchap4という新しいディレクトリにコピーし、新しく作成したディレクトリに移動します。

Windowsの場合:

```
mkdir c:\program files\Apache Group\Apache2\htdocs\chap4
cd c:\program files\Apache Group\Apache2\htdocs\chap4
copy ..\chap3\*.*
```

Linuxの場合:

```
mkdir $HOME/public_html/chap4
cd $HOME/public_html/chap4
cp ../chap3/*.
```

2. 任意のエディタを使用して、anycos_cn.incというファイルを作成し、データベース接続情報の名前付き定数を定義します。このファイルを使用すると、接続情報を1箇所ですべて変更できます。

```
<?php // File: anycos_cn.inc

define('ORA_CON_UN', 'hr'); // User name
define('ORA_CON_PW', 'hr'); // Password
define('ORA_CON_DB', 'localhost/orcl'); // Connection identifier

?>
```

ここでは、わかりやすくするために、サンプル・アプリケーション・コードにユーザー名およびパスワードが書き込まれています。デプロイするアプリケーションでは、アプリケーションのソース・コードにユーザー名およびパスワードの文字列を直接コーディングしないことをお勧めします。ユーザーにユーザー名およびパスワードの入力を求めるダイアログ・ボックスを実装する方法などのより安全な方法を使用することをお勧めします。

セキュリティ機能およびセキュリティ・プラクティスの詳細は、『Oracle Databaseセキュリティ・ガイド』および開発環境のドキュメントを参照してください。

3. データベース接続の作成、問合せの実行およびデータベースからの切断を実行する関数を宣言するanycos_db.incというファイルを作成します。db_error()という追加の関数をコールすることによって管理されるエラー処理を含む次のロジックを使用します。

```

<?php // File: anyco_db.inc

function db_connect()
{
    // use constants defined in anyco_cn.inc
    $conn = oci_connect(ORA_CON_UN, ORA_CON_PW, ORA_CON_DB);
    if (!$conn) {
        db_error(null, __FILE__, __LINE__);
    }
    return($conn);
}

function db_do_query($conn, $statement)
{
    $stid = oci_parse($conn, $statement);
    if (!$stid) {
        db_error($conn, __FILE__, __LINE__);
    }

    $r = oci_execute($stid, OCI_DEFAULT);
    if (!$r) {
        db_error($stid, __FILE__, __LINE__);
    }
    $r = oci_fetch_all($stid, $results, null, null,
        OCI_FETCHSTATEMENT_BY_ROW);
    return($results);
}

// $r is the resource containing the error.
// Pass no argument or false for connection errors

function db_error($r = false, $file, $line)
{
    $err = $r ? oci_error($r) : oci_error();

    if (isset($err['message'])) {
        $m = htmlentities($err['message']);
    }
    else {
        $m = 'Unknown DB error';
    }

    echo '<p><b>Error</b>: at line '.$line.' of '.$file.'</p>';
    echo '<pre>'.$m.'</pre>';

    exit;
}

?>

```

この例のdb_do_query()関数では、oci_fetch_all() OCI8関数を使用します。oci_fetch_all()関数は、次の5つのパラメータをとります。

- \$stid (実行する文の文識別子)。
- \$results (問合せで戻されたデータが格納される出力配列変数)。
- 3つ目のパラメータは最初にスキップする行数を示し、NULLの場合は無視されます。
- 4つ目のパラメータはフェッチする最大行数を示し、NULLの場合は無視されます。この場合、問合せに該当するすべての行が戻されます。この例では、結果セットが大きくないため、それでも問題ありません。
- 最後のパラメータ・フラグOCI_FETCHSTATEMENT_BY_ROWは、\$results配列内のデータが行ごとに編成され、各行に列値の配列が含まれていることを示します。値がOCI_FETCHSTATEMENT_BY_COLUMNの場合、\$results配列は列ごとに編成され、各列エントリに各行の列値の配列が含まれます。このフラグに選択する値は、ロジックでデータを処理する方法によって異なります。

結果配列の構造を調べるには、問合せの実行後、PHPのvar_dump()関数を使用します。これはデバッグに有効です。次に例を示します。

```
print '<pre>';
var_dump($results);
print '</pre>';
```

db_error()関数は、3つの引数をとります。\$rパラメータには、接続エラーを取得する場合はfalseまたはNULLを指定し、接続リソースまたは文リソースに関連するエラーを取得する場合はそのリソースを指定できます。\$file値および\$line値には__FILE__および__LINE__を実際のパラメータとして使用してそれぞれ値が移入されるため、データベース・エラーのレポート元のソース・ファイルおよび行をエラー・メッセージに表示できます。これによって、エラーの原因を簡単に追跡できるようになります。

db_error()関数は、oci_error()関数をコールしてデータベース・エラー・メッセージを取得します。

db_error()関数は、メッセージを出力する前にisset()関数をコールします。isset()関数は、データベース・エラー構造のメッセージ・コンポーネントが設定されているかどうか、またはエラーが不明なものであるかどうかを確認します。

4. anyco_ui.incを編集します。DEPARTMENTS表の問合せから1行取得した結果をHTML表形式に設定するには、次の関数を挿入します。

```
function ui_print_department($dept)
{
    if (!$dept) {
        echo '<p>No Department found</p>';
    }
    else {
        echo <<<END
        <table>
        <tr>
        <th>Department<br>ID</th>
        <th>Department<br>Name</th>
        <th>Manager<br>Id</th>
        <th>Location ID</th>
        </tr>
        <tr>
        END;
        echo '<td>'.htmlentities($dept['DEPARTMENT_ID']).</td>';
        echo '<td>'.htmlentities($dept['DEPARTMENT_NAME']).</td>';
        echo '<td>'.htmlentities($dept['MANAGER_ID']).</td>';
        echo '<td>'.htmlentities($dept['LOCATION_ID']).</td>';
        echo <<<END
        </tr>
        </table>
        END;
    }
}
```

第3章で説明したように、END;行の先頭には空白を配置しないでください。配置すると、残りのドキュメントが、出力対象テキストの一部として処理されます。

5. anyco.phpファイルを編集します。anyco_ui.incおよびanyco_db.incファイルをインクルードし、次のコードを使用して、department_idが80の部門の情報を問い合わせて表示するデータベース関数をコールします。ファイルは、次のようになります。

```
<?php // File: anyco.php

require('anyco_cn.inc');
require('anyco_db.inc');
require('anyco_ui.inc');

$query =
'SELECT department_id, department_name, manager_id, location_id
FROM departments
WHERE department_id = 80';

$conn = db_connect();

$dept = db_do_query($conn, $query);
ui_print_header('Departments');
ui_print_department($dept[0]);
ui_print_footer(date('Y-m-d H:i:s'));

?>
```

6. アプリケーションに対して行った変更をテストするには、ブラウザのウィンドウに次のURLを入力します。

Windowsの場合:

`http://localhost/chap4/anyco.php`

Linuxの場合:

`http://localhost/~<username>/chap4/anyco.php`

ブラウザ・ウィンドウに、次のようなページが戻されます。

Department ID	Department Name	Manager Id	Location ID
80	Sales	145	2500

2005-09-30 11:50:00 Any Co.

[chap4_db_connect_002.gifの説明](#)

バインド変数を使用した問合せの作成

WHERE句に値を含めた問合せを使用すると有効な場合があります。ただし、問合せの条件値が変更される可能性が高い場合は、問合せに値をエンコードするのは適切ではありません。バインド変数を使用することをお勧めします。

バインド変数は、問合せ内でコロンの後に指定するシンボリック名で、リテラル値のプレースホルダとして動作します。たとえば、`anyco.php` ファイルに作成した問合せ文字列は、バインド変数:`did`を使用して次のように記述しなおすことができます。

```
$query =  
'SELECT department_id, department_name, manager_id, location_id  
FROM departments  
WHERE department_id = :did';
```

バインド変数を使用してSQL文をパラメータ化すると、次のメリットがあります。

- コードを変更せずに様々な入力値で文を再利用できます。
- Oracle Databaseでは、以前と同じ問合せ文字列を呼出したときの解析情報を再利用できるため、サーバーの問合せ解析時間が短縮され、問合せのパフォーマンスが向上します。
- SQLインジェクションというセキュリティの問題に対する保護機能があります。
- ユーザー入力で、引用符を特別に処理する必要はありません。

問合せでバインド変数を使用する場合は、問合せを実行する前に、PHPコードで、問合せで使用している各バインド変数（プレースホルダ）に実際の値を関連付ける必要があります。このプロセスは、ランタイム・バインディングと呼ばれます。

PHPアプリケーションで問合せのバインド変数を使用できるようにするには、PHPアプリケーション・コードに対して次の変更を行います。

1. anyco.phpファイルを編集します。 バインド変数が使用されるように問合せを変更し、バインド変数に関連付ける値を格納するための配列を作成して、db_do_query()関数に\$bindargs配列を渡します。

```
<?php // File: anyco.php
...

$query =
'SELECT department_id, department_name, manager_id, location_id
FROM departments
WHERE department_id = :did';

$bindargs = array();
// In the $bindargs array add an array containing
// the bind variable name used in the query, its value, a length
array_push($bindargs, array('DID', 80, -1));

$conn = db_connect();
$dept = db_do_query($conn, $query, $bindargs);

...
?>
```

この例では、DIDというバインド変数は、パラメータ化した問合せの入力引数であり、値80に関連付けられています。このバインド変数の値は、後で動的に決定されます。また、OCI8レイヤーで長さを決定できるように、長さ構成要素を-1として渡します。バインド変数を使用してデータベースから出力を戻す場合は、サイズを明示的に指定する必要があります。

2. anyco_db.incファイルを編集します。\$bindvars配列変数を3つ目のパラメータとしてとるようにdb_do_query()関数を変更します。oci_bind_by_name() OCI8コールをコールして、\$bindvarsパラメータに指定したPHP値を問合せのバインド変数に関連付けます。ファイルは、次のようになります。

```
function db_do_query($conn, $statement, $bindvars = array())
{
    $stid = oci_parse($conn, $statement);
    if (!$stid) {
        db_error($conn, __FILE__, __LINE__);
    }

    // Bind the PHP values to query bind parameters
    foreach ($bindvars as $b) {
        // create local variable with caller specified bind value
        $$b[0] = $b[1];
        // oci_bind_by_name(resource, bv_name, php_variable, length)
        $r = oci_bind_by_name($stid, ":".$b[0], $$b[0], $b[2]);
        if (!$r) {
            db_error($stid, __FILE__, __LINE__);
        }
    }
    $r = oci_execute($stid, OCI_DEFAULT);
    if (!$r) {
        db_error($stid, __FILE__, __LINE__);
    }
    $r = oci_fetch_all($stid, $results, null, null,
        OCI_FETCHSTATEMENT_BY_ROW);
    return($results);
}
```

バインドは、oci_execute()が実行される前にforeachループで実行されます。

\$bindvars配列の各エンタリでは、最初の要素に問合せのバインド変数名が含まれており、この名前を使用して同じ名前のPHP変数が作成されます。つまり、\$\$b[0]は\$b[0]の値DIDをとり、\$DIDというPHP変数を作成します。このPHP変数の値は、エンタリの2つ目の要素から割り当てられます。

oci_bind_by_name()関数は、4つのパラメータをとります。リソースとしての\$stid、配列エンタリの最初の要素から導出された問合せ内のバインド変数名を表す文字列、そのバインド変数に関連付けられる値が格納されているPHP変数、および入力値の長さの4つです。

3. ここまでの変更の結果をテストするには、anyco.phpファイルおよびanyco_db.incファイルを保存し、次のURLを入力します。

Windowsの場合:

http://localhost/chap4/anyco.php

Linuxの場合:

http://localhost/~<username>/chap4/anyco.php

ブラウザ・ウィンドウに、次のようなページが戻されます。

Departments			
Department ID	Department Name	Manager Id	Location ID
80	Sales	145	2500

2005-09-30 14:42:50 Any Co.

[chap4_db_connect_003.gifの説明](#)

データベース・レコードのナビゲート

データベース・レコードのナビゲーションを追加するには、アプリケーション・ロジックに対していくつかの重要な変更を行う必要があります。この変更を行うには、次の操作を組み合わせる必要があります。

- データベース・レコードを1つずつ移動する「Next」および「Previous」ナビゲーション・ボタンを提供するHTMLフォームをインクルードします。
- ページのHTTPリクエストが、「Next」または「Previous」ボタンをクリックしてポストされたかどうかを検出します。
- HTTPセッション・ステートを使用して最後に問い合わせた行を追跡します。HTTPリクエスト間で特定のクライアントの状態情報を保持するためにPHPセッションを開始します。最初のHTTPリクエストが最初のデータ行を取り出し、セッション・ステートを初期化します。ナビゲーション・ボタンで開始した後続リクエストでは、前のHTTPリクエストのセッション・ステートが組み合わせられるため、問合せで次に取り出すレコードを制御する変数をアプリケーションに設定できます。
- アプリケーションの状態によって値が決まる一連の条件に基づいて、行のサブセットを戻す問合せを作成します。

データベース行のナビゲーションを追加するには、次の手順を実行します。

1. anyco_ui.incファイルを編集します。「Departments」ページに「Next」および「Previous」ナビゲーション・ボタンを追加します。ui_print_department()関数に、フォーム属性actionの値を指定する\$posturl1という2つ目のパラメータを追加します。</table>タグを出力した後、「Next」および「Previous」ボタンのHTMLフォーム・タグをインクルードします。

```
<?php // File: anyco_ui.inc
...
function ui_print_department($dept, $posturl)
{
    ...
    echo <<<END
</tr>
</table>
<form method="post" action="$posturl">
<input type="submit" value="< Previous" name="prevdept">
<input type="submit" value="Next >" name="nextdept">
</form>
```

```

END;
}
}

?>

```

2. anyco.phpファイルを編集します。「Next」または「Previous」ボタンを使用してページを起動したかどうかを検出し、セッション・ステートを追跡するには、PHP関数session_start()をコールし、construct_departments()という関数を作成します。

データベース・アクセス・ロジックを新しいconstruct_departments()関数に移動して変更します。この関数は、ナビゲーションが実行されたかどうかを検出し、セッション・ステートを管理し、処理するデータベース・アクセス・レイヤーの副問合せを定義し、db_get_page_data()関数に接続してコールします。ファイルは、次のようになります。

```

<?php // File: anyco.php

require('anyco_cn.inc');
require('anyco_db.inc');
require('anyco_ui.inc');

session_start();
construct_departments();

function construct_departments()
{
    if (isset($_SESSION['currentdept']) &&
        isset($_POST['prevdept']) &&
        $_SESSION['currentdept'] > 1) {
        $current = $_SESSION['currentdept'] - 1;
    }
    elseif (isset($_SESSION['currentdept']) &&
            isset($_POST['nextdept'])) {
        $current = $_SESSION['currentdept'] + 1;
    }
    elseif (isset($_POST['showdept']) &&
            isset($_SESSION['currentdept'])) {
        $current = $_SESSION['currentdept'];
    }
    else {
        $current = 1;
    }

    $query = 'SELECT department_id, department_name,
              manager_id, location_id
              FROM departments
              ORDER BY department_id asc';

    $conn = db_connect();

    $dept = db_get_page_data($conn, $query, $current, 1);
    $deptid = $dept[0]['DEPARTMENT_ID'];

    $_SESSION['currentdept'] = $current;

    ui_print_header('Department');
    ui_print_department($dept[0], $_SERVER['SCRIPT_NAME']);
    ui_print_footer(date('Y-m-d H:i:s'));
}

?>

```

construct_departments()関数の先頭にあるifおよびelseif構文は、HTTPポスト・リクエストでナビゲーション・ボタンを使用してページを処理したかどうかを検出し、セッション・ステートにcurrentdeptの数値が設定されているかどうかを追跡するために使用されています。変数\$currentは状況によって異なり、「Previous」ボタンをクリックすると1つ減分され、「Next」ボタンをクリックすると1つ増分されます。それ以外の場合は、現行の部門に設定されるか、または初めて処理に対して1に初期化されます。

department_idのすべての部門行を昇順で取得するように問合せが形成されています。ORDER BY句は、ナビゲーション・ロジックに不可欠な部分です。この問合せは、多くの行で構成されているページを取得するdb_get_page_data()関数内で副問合せとして使用されています。取得するページのページ当たりの行数は、db_get_page_data()関数の4つ目の引数として指定されています。データベースに接続した後、指定した問合せで取得した行セットを取り出すためにdb_get_page_data()がコールされています。db_get_page_data()関数には、接続リソース、問合せ文字列、必要なデータ行の次ページの最初の行を示す\$currentの値およびページ当たりの行数（この例では1ページ当たり1行）が指定されています。

1ページ分の行を取得するためにdb_get_page_data()がコールされた後、\$current()の値がアプリケーションのセッション・ス

データに保存されています。

ページのヘッダーとフッターを出力する間に、最近フェッチした部門行を表示するために`ui_print_department()`関数がコールされています。`ui_print_department()`関数は、`$_SERVER['SCRIPT_NAME']`を使用して、`$posturl`パラメータに現行のPHPスクリプト名を指定します。これで、HTMLフォームにアクション属性が設定され、「Next」または「Previous」ボタンをクリックするたびに、`anyco.php`ファイルがコールされます。

3. `anyco_db.inc`ファイルを編集します。行のサブセットを問い合わせるように`db_get_page_data()`関数を実装します。

```
// Return subset of records
function db_get_page_data($conn, $q1, $current = 1,
    $rowsperpage = 1, $bindvars = array())
{
    // This query wraps the supplied query, and is used
    // to retrieve a subset of rows from $q1
    $query = 'SELECT *
        FROM (SELECT A.*, ROWNUM AS RNUM
            FROM (.' . $q1 . ') A
            WHERE ROWNUM <= :LAST)
        WHERE :FIRST <= RNUM';

    // Set up bind variables.
    array_push($bindvars, array('FIRST', $current, -1));
    array_push($bindvars,
        array('LAST', $current+$rowsperpage-1, -1));

    $r = db_do_query($conn, $query, $bindvars);
    return($r);
}
```

`db_get_page_data()`関数で問合せの構造を使用すると、一連の（1ページ分の）データベース行をナビゲートできます。

`$q1`に指定した問合せは、次の副問合せ内に副問合せとしてネストされています。

```
SELECT A. *, ROWNUM AS RNUM FROM $q1 WHERE ROWNUM <= :LAST
```

`$q1`に指定した問合せは、順序づけられた一連の行を取り出します。これらの行は、最初の行から次ページのサイズ（`$rowsperpage`）までのすべての行を戻す囲み問合せでフィルタ処理されています。OracleのROWNUM関数（または疑似列）が、`$q1`に指定した問合せによって戻される各行に対して1で始まる整数を戻すため、この操作を実行できます。

副問合せを囲む問合せ`$q1`によって戻される行セットは、次の最も外側の問合せの条件で再度フィルタ処理されています。

```
WHERE :FIRST <= RNUM
```

この条件によって、`:FIRST`の値（`$current`の値）より前にある行が最終的な行セットから除外されます。この問合せを使用すると、最初の行が`$current`値で決まり、ページ・サイズが`$rowsperpage`値で決まる行セットをナビゲートできます。

`$current`値は、`:FIRST`というバインド変数に関連付けられています。式`$current+$rowsperpage-1`によって、`:LAST`バインド変数に関連付けられている値が設定されます。

4. アプリケーションに対して行った変更をテストするには、変更後のファイルを保存し、Webブラウザに次のURLを入力します。

Windowsの場合:

```
http://localhost/chap4/anyco.php
```

Linuxの場合:

```
http://localhost/~<username>/chap4/anyco.php
```

`anyco.php`ページをリクエストすると、DEPARTMENT表の最初のレコードである「Administration」部門が表示されます。

Department

Department ID	Department Name	Manager Id	Location ID
10	Administration	200	1700

< Previous Next >

2005-10-02 22:58:19 Any Co.

[chap4_db_nagivation_001.gifの説明](#)

5. 次の部門レコード（「Marketing」）にナビゲートするには、「Next」をクリックします。

Department

Department ID	Department Name	Manager Id	Location ID
20	Marketing	201	1800

< Previous Next >

2005-10-02 22:59:10 Any Co.

[chap4_db_nagivation_002.gifの説明](#)

6. 最初の部門レコード（「Administration」）に戻るには、「Previous」をクリックします。

Department

Department ID	Department Name	Manager Id	Location ID
10	Administration	200	1700

< Previous Next >

2005-10-02 22:59:29 Any Co.

[chap4_db_nagivation_003.gifの説明](#)

必要に応じて、「Next」および「Previous」をクリックしてDEPARTMENTS表の他のレコードにナビゲートして、アプリケーションのテストおよび試用を継続して行うことができます。

注意:

DEPARTMENTS表の最後のレコードを越えてナビゲートすると、エラーが発生します。エラー処理の詳細は、第5章の「[エラー・リカバリの追加](#)」を参照してください。

ROWNUMとROW_NUMBER()

ハードコードされた問合せでPHP関数を記述する場合、戻される行数を制限する方法としてはROW_NUMBER()関数の方が簡単なことがあります。たとえば、すべての従業員の姓を戻す問合せがあるとします。

```
SELECT last_name FROM employees ORDER BY last_name;
```

次のように記述すると、51～100までの行を選択できます。

```
SELECT last_name FROM  
SELECT last_name, ROW_NUMBER() OVER (ORDER BY last_name R FROM employees)  
where R BETWEEN 51 AND 100;
```

基本的な「Departments」ページの拡張

次の追加情報が含まれるように、「Departments」ページを拡張します。

- 部門のマネージャの名前
- 部門に配属されている従業員の数
- 部門の場所を識別する国名

これらの追加情報は、DEPARTMENTS、EMPLOYEES、LOCATIONS、COUNTRIESの各表間で結合操作を実行するように問合せを変更することによって取得できます。

「Departments」ページを拡張するには、次の手順を実行します。

1. anyco_ui.incファイルを編集します。「Manager ID」および「Location ID」への参照をそれぞれ「Manager Name」および「Location」に置き換え、「Department Name」の後に「Number of Employees」フィールドを挿入して、ui_print_department()関数を変更します。表のヘッダーおよびデータ・フィールドに必要な変更を行います。ファイルは、次のようになります。

```
function ui_print_department($dept, $posturl)
{
    if (!$dept) {
        echo '<p>No Department found</p>';
    }
    else {
        echo <<<END
        <table>
        <tr>
        <th>Department<br>ID</th>
        <th>Department<br>Name</th>
        <th>Number of<br>Employees</th>
        <th>Manager<br>Name</th>
        <th>Location</th>
        </tr>
        <tr>
        END;
        echo '<td>'.htmlentities($dept['DEPARTMENT_ID']).'</td>';
        echo '<td>'.htmlentities($dept['DEPARTMENT_NAME']).'</td>';
        echo '<td>'.htmlentities($dept['NUMBER_OF_EMPLOYEES']).'</td>';
        echo '<td>'.htmlentities($dept['MANAGER_NAME']).'</td>';
        echo '<td>'.htmlentities($dept['COUNTRY_NAME']).'</td>';
        echo <<<END
        </tr>
        </table>
        <form method="post" action="$posturl">
```

```

<input type="submit" value="< Previous" name="prevdept">
<input type="submit" value="Next >" name="nextdept">
</form>
END;
}
}

```

2. anyco.phpファイルを編集します。construct_departments()の問合せ文字列を次のように置き換えます。

```

$query =
"SELECT d.department_id, d.department_name,
       substr(e.first_name,1,1)||'. '|| e.last_name as manager_name,
       c.country_name, count(e2.employee_id) as number_of_employees
FROM   departments d, employees e, locations l,
       countries c, employees e2
WHERE  d.manager_id = e.employee_id
AND    d.location_id = l.location_id
AND    d.department_id = e2.department_id
AND    l.country_id = c.country_id
GROUP BY d.department_id, d.department_name,
         substr(e.first_name,1,1)||'. '||e.last_name,
         c.country_name
ORDER BY d.department_id ASC";

```

この文ではSQLリテラル文字列が一重引用符で囲まれているため、文の記述を簡略化するために問合せ文字列は二重引用符で囲まれています。

3. ファイルへの変更を保存し、Webブラウザに次のURLを入力して、変更をテストします。

Windowsの場合:

<http://localhost/chap4/anyco.php>

Linuxの場合:

<http://localhost/~<username>/chap4/anyco.php>

Webページの結果は、次のような出力になります。

Department				
Department ID	Department Name	Number of Employees	Manager Name	Location
10	Administration	4	J. Whalen	United States of America
<input style="border: none;" type="button" value=" < Previous "/> <input style="border: none;" type="button" value=" Next > "/>				
2005-10-03 10:56:55				Any Co.

[chap4_enhance_dept_001.gifの説明](#)

5 データの更新

この章では、従業員レコードを挿入、更新および削除できるフォームを使用してAnyco HRアプリケーションを拡張します。

- [基本的な「Employees」ページの構築](#)
- [基本的な「Employees」ページの拡張](#)
- [「Departments」と「Employees」の結合](#)
- [エラー・リカバリの追加](#)
- [追加のエラー処理](#)

基本的な「Employees」ページの構築

この項では、基本的な「Employees」ページが含まれるようにアプリケーションを拡張します。

従業員レコードを表示するには、次の手順を実行します。

1. chap5ディレクトリを作成し、chap4からアプリケーション・ファイルをコピーし、新しく作成したディレクトリに移動します。

Windowsの場合:

```
mkdir c:\program files\Apache Group\Apache2\htdocs\chap5
cd c:\program files\Apache Group\Apache2\htdocs\chap5
copy ..\chap4\* .
```

Linuxの場合:

```
mkdir $HOME/public_html/chap5
cd $HOME/public_html/chap5
cp ../chap4/* .
```

2. anyco.phpファイルを編集します。construct_employees()関数を追加します。この関数は、従業員問合せを作成し、db_do_query()関数をコールしてその問合せを実行し、ui_print_employees()関数を使用してその結果を出力します。

```
function construct_employees()
{
    $query =
    "SELECT employee_id,
       substr(first_name,1,1) || ' ' || last_name as employee_name,
       hire_date,
       to_char(salary, '9999G999D99') as salary,
       nvl(commission_pct,0) as commission_pct
    FROM employees
    ORDER BY employee_id asc";

    $conn = db_connect();
    $emp = db_do_query($conn, $query);

    ui_print_header('Employees');
    ui_print_employees($emp);
    ui_print_footer(date('Y-m-d H:i:s'));
}
```

この問合せではバインド変数を使用しないため、db_do_query()コールに\$bindargsパラメータを渡す必要はありません。

db_do_query()を宣言すると、デフォルト値（空配列）が自動的に指定されます。PHPでは、関数に様々な数のパラメータを含めることができます。

3. anyco.phpファイルを編集します。construct_departments()へのコールをconstruct_employees()へのコールに置き換えます。

```
<?php // File: anyco.php

require('anyco_cn.inc');
require('anyco_db.inc');
require('anyco_ui.inc');

session_start();
construct_employees();
...
?>
```

4. anyco_ui.incファイルを編集します。ui_print_employees()関数を追加して、HTML表の従業員データの表示を実装します。

```
function ui_print_employees($employee records)
{
    if (!$employee records) {
        echo '<p>No Employee found</p>';
    }
    else {
        echo <<<END
        <table>
        <tr>
        <th>Employee<br>ID</th>
        <th>Employee<br>Name</th>
        <th>Hiredate</th>
        <th>Salary</th>
        <th>Commission<br>(%)</th>
        </tr>
        END;
        // Write one row per employee
        foreach ($employee records as $emp) {
            echo '<tr>';
            echo '<td align="right">'.
                htmlentities($emp['EMPLOYEE_ID']).</td>';
            echo '<td>'.htmlentities($emp['EMPLOYEE_NAME']).</td>';
            echo '<td>'.htmlentities($emp['HIRE_DATE']).</td>';
            echo '<td align="right">'.
                htmlentities($emp['SALARY']).</td>';
            echo '<td align="right">'.
                htmlentities($emp['COMMISSION_PCT']).</td>';
            echo '</tr>';
        }
        echo <<<END
        </table>
        END;
    }
}
```

5. anyco.phpファイルおよびanyco_ui.incファイルへの変更を保存します。Webブラウザに次のURLを入力して、これらの変更の結果をテストします。

Windowsの場合:

<http://localhost/chap5/anyco.php>

Linuxの場合:

<http://localhost/~<username>/chap5/anyco.php>

結果ページを調べ、下へスクロールして、ページに表示されているすべての従業員レコードを参照します。

Employees

Employee ID	Employee Name	Hiredate	Salary	Commission (%)
100	S. King	17-JUN-87	24,000.00	0
101	N. Kochhar	21-SEP-89	17,000.00	0
102	L. De Haan	13-JAN-93	17,000.00	0
103	A. Hunold	03-JAN-90	9,000.00	0
104	B. Ernst	21-MAY-91	6,000.00	0
105	D. Austin	25-JUN-97	4,800.00	0
106	V. Pataballa	05-FEB-98	4,800.00	0
107	D. Lorentz	07-FEB-99	4,200.00	0
108	N. Greenberg	17-AUG-94	12,000.00	0

[chap5_basic_emp_001.gifの説明](#)

基本的な「Employees」ページの拡張

この項では、従業員レコードを操作できるように基本的な「Employees」ページを拡張します。

従業員レコードを操作できるようにするには、次の手順を実行します。

1. anyco.phpファイルを編集します。construct_employees()コールを、従業員レコードの表示、挿入、更新および削除のリクエストを管理するフォーム・ハンドラ制御ロジックに置き換えます。

```
<?php // File: anyco.php

require('anyco_cn.inc');
require('anyco_db.inc');
require('anyco_ui.inc');

session_start();
// Start form handler code
if (isset($_POST['insertemp'])) {
    construct_insert_emp();
}
elseif (isset($_POST['saveinsertemp'])) {
    insert_new_emp();
}
elseif (isset($_POST['modifyemp'])) {
    construct_modify_emp();
}
elseif (isset($_POST['savemodifiedemp'])) {
    modify_emp();
}
elseif (isset($_POST['deleteemp'])) {
    delete_emp();
}
else {
    construct_employees();
}

...

```

2. anyco.phpファイルを編集します。 construct_insert_emp()関数を追加します。

```
function construct_insert_emp()
{
    $conn = db_connect();

    $query = "SELECT job_id, job_title
              FROM jobs
              ORDER BY job_title ASC";
    $jobs = db_do_query($conn, $query,
                      OCI_FETCHSTATEMENT_BY_COLUMN);

    $query = "SELECT sysdate FROM dual";
    $date = db_do_query($conn, $query,
                      OCI_FETCHSTATEMENT_BY_COLUMN);
    $emp = array(
        'DEPARTMENT_ID' => 10, // Default to department 10
        'HIRE_DATE' => $date['SYSDATE'][0],
        'ALLJOBIDS' => $jobs['JOB_ID'],
        'ALLJOBTTLES' => $jobs['JOB_TITLE']
    );

    ui_print_header("Insert New Employee");
    ui_print_insert_employee($emp, $_SERVER['SCRIPT_NAME']);
    // Note: The two kinds of date used:
    // 1) SYSDATE for current date of the database system, and
    // 2) The PHP date for display in the footer of each page
    ui_print_footer(date("Y-m-d H:i:s"));
}
```

construct_insert_emp()関数は、2つの問合せを実行して、「Insert New Employee」フォームへの移入に使用するデフォルト・データを取得します。このフォームは、ui_print_insert_employee()関数によって表示されます。

JOBS表の\$queryは、既存のすべてのジョブIDおよびその説明のリストを取得して、ui_print_insert_employee()関数によって生成されたHTMLフォームにジョブ・タイプ選択用のリストを作成します。

SYSDATEが使用されている\$queryは、新しい従業員のデフォルトの 用日を設定するために、現在のデータベース日時を取得します。

このアプリケーション・コードでは、2種類の日付が使用されています。PHPのdate()関数は、ページ・フッターに日時を出力します。OracleのSYSDATE関数は、「Employees」ページの 用日フィールドに表示するデフォルトの日時を取得し、テキストが正しいデータベース形式で入力されるようにします。

2つのdb_do_query()関数コールには、問合せの戻り が列値の配列になるように指定する追加のパラメータ値 OCI_FETCHSTATEMENT_BY_COLUMNが含まれています。

3. anyco.phpファイルを編集します。 insert_new_emp()関数を追加して、EMPLOYEES表に従業員レコードを挿入します。

```
function insert_new_emp()
{
    $newemp = $_POST;
    $statement =
        "INSERT INTO employees
        (employee_id, first_name, last_name, email, hire_date,
         job_id, salary, commission_pct, department_id)
        VALUES (employees_seq.nextval, :fnm, :lnm, :eml, :hdt, :jid,
                :sal, :cpt, :did)";

    $conn = db_connect();
    $emailid = $newemp['firstname'].$newemp['lastname'];

    $bindargs = array();
    array_push($bindargs, array('FNM', $newemp['firstname'], -1));
    array_push($bindargs, array('LNM', $newemp['lastname'], -1));
    array_push($bindargs, array('EML', $emailid, -1));
    array_push($bindargs, array('HDT', $newemp['hiredate'], -1));
    array_push($bindargs, array('JID', $newemp['jobid'], -1));
    array_push($bindargs, array('SAL', $newemp['salary'], -1));
    array_push($bindargs, array('CPT', $newemp['commpct'], -1));
    array_push($bindargs, array('DID', $newemp['deptid'], -1));
}
```

```

$r = db_execute_statement($conn, $statement, $bindargs);
construct_employees();
}

```

db_execute_statement()関数の戻り値は無視され、変数にも割り当てられません。関数の結果に対してアクションが実行されないためです。

- anyco.phpファイルを編集します。construct_modify_emp()関数を追加して、従業員レコードを更新するためのHTMLフォームを構築します。

```

function construct_modify_emp()
{
    $empid = $_POST['emprec'];
    $query =
        "SELECT employee_id, first_name, last_name, email, hire_date,
           salary, nvl(commission_pct,0) as commission_pct
        FROM employees
        WHERE employee_id = :empid";

    $conn = db_connect();
    $bindargs = array();
    array_push($bindargs, array('EMPID', $empid, -1));

    $emp = db_do_query($conn, $query, OCI_FETCHSTATEMENT_BY_ROW,
                      $bindargs);

    ui_print_header('Modify Employee ');
    ui_print_modify_employee($emp[0], $_SERVER['SCRIPT_NAME']);
    ui_print_footer(date("Y-m-d H:i:s"));
}

```

- anyco.phpファイルを編集します。modify_emp()関数を追加して、更新フォーム・フィールドの値でEMPLOYEES表の従業員レコードを更新します。

```

function modify_emp()
{
    $newemp = $_POST;
    $statement =
        "UPDATE employees
        SET first_name = :fnm, last_name = :lnm, email = :eml,
            salary = :sal, commission_pct = :cpt
        WHERE employee_id = :eid";

    $conn = db_connect();
    $bindargs = array();
    array_push($bindargs, array('EID', $newemp['empid'], -1));
    array_push($bindargs, array('FNM', $newemp['firstname'], -1));
    array_push($bindargs, array('LNM', $newemp['lastname'], -1));
    array_push($bindargs, array('EML', $newemp['email'], -1));
    array_push($bindargs, array('SAL', $newemp['salary'], -1));
    array_push($bindargs, array('CPT', $newemp['commpct'], -1));

    $r = db_execute_statement($conn, $statement, $bindargs);
    construct_employees();
}

```

- anyco.phpファイルを編集します。delete_emp()関数を追加して、EMPLOYEES表から従業員レコードを削除します。

```

function delete_emp()
{
    $empid = $_POST['emprec'];
    $statement = "DELETE FROM employees
                WHERE employee_id = :empid";
    $conn = db_connect();
    $bindargs = array();
    array_push($bindargs, array('EMPID', $empid, 10));
    $r = db_execute_statement($conn, $statement, $bindargs);

    construct_employees();
}

```

- anyco.phpファイルを編集します。construct_employees()関数で、db_do_query()コールの最後のパラメータとして

OCI_FETCHSTATEMENT_BY_ROWを指定し、ui_print_employees()コールの2つ目のパラメータとして\$_SERVER['SCRIPT_NAME']を指定します。ファイルは、次のようになります。

```
function construct_employees()
{
    $query =
        "SELECT employee_id,
           substr(first_name,1,1) || '. ' || last_name as employee_name,
           hire_date,
           to_char(salary, '9999G999D99') as salary,
           nvl(commission_pct,0) as commission_pct
        FROM employees
        ORDER BY employee_id asc";

    $conn = db_connect();
    $emp = db_do_query($conn, $query, OCI_FETCHSTATEMENT_BY_ROW);

    ui_print_header('Employees');
    ui_print_employees($emp, $_SERVER['SCRIPT_NAME']);
    ui_print_footer(date('Y-m-d H:i:s'));
}
```

8. anyco_db.incファイルを編集します。db_do_query()関数に3つ目のパラメータとして\$resulttypeを追加します。コール元が出力タイプを選択できるように、oci_fetch_all()コールの最後のパラメータ値OCI_FETCHSTATEMENT_BY_ROWを変数に置き換えます。

```
function db_do_query($conn, $statement, $resulttype,
                    $bindvars = array())
{
    $stid = oci_parse($conn, $statement);
    ...

    $r = oci_fetch_all($stid, $results, null, null, $resulttype);
    return($results);
}
```

9. anyco_db.incファイルを編集します。db_get_page_data()関数内で、db_do_query()コールの3つ目のパラメータ値としてOCI_FETCHSTATEMENT_BY_ROWを挿入します。

```
function db_get_page_data($conn, $ql, $current = 1,
                          $rowsperpage = 1, $bindvars = array())
{
    ...

    $r = db_do_query($conn, $query, OCI_FETCHSTATEMENT_BY_ROW, $bindvars);
    return($r);
}
```

10. anyco_db.incファイルを編集します。db_execute_statement()関数を追加して、INSERT文などデータ操作文を実行します。

```
function db_execute_statement($conn, $statement, $bindvars = array())
{
    $stid = oci_parse($conn, $statement);
    if (!$stid) {
        db_error($conn, __FILE__, __LINE__);
    }
    // Bind parameters
    foreach ($bindvars as $b) {
        // create local variable with caller specified bind value
        $$b[0] = $b[1];
        $r = oci_bind_by_name($stid, ":$b[0]", $$b[0], $b[2]);
        if (!$r) {
            db_error($stid, __FILE__, __LINE__);
        }
    }

    $r = oci_execute($stid);
    if (!$r) {
        db_error($stid, __FILE__, __LINE__);
    }
}
```



```

<td>First Name</td>
<td><input type="text" name="firstname" size="20"></td>
</tr>
<tr>
<td>Last Name</td>
<td><input type="text" name="lastname" size="20"></td>
</tr>
<tr>
<td>Hiredate</td>
<td><input type="text" name="hiredate" value="$hiredate"
size="20"></td>
</tr>
<tr>
<td>Job</td>
<td><select name="jobid">
END;
// Write the list of jobs
for ($i = 0; $i < count($emp['ALLJOBIDS']); $i++)
{
echo '<option
label="'.htmlentities($emp['ALLJOBTTLES'][$i])."'
value="'.htmlentities($emp['ALLJOBIDS'][$i])."'>'.
htmlentities($emp['ALLJOBTTLES'][$i]).'</option>';
}
echo <<<END
</select>
</td>
</tr>
<tr>
<td>Salary</td>
<td><input type="text" name="salary" value="1"
size="20"></td>
</tr>
<tr>
<td>Commission (%)</td>
<td><input type="text" name="commpct" value="0"
size="20"></td>
</tr>
</table>
<input type="submit" value="Save" name="saveinsertemp">
<input type="submit" value="Cancel" name="cancel">
</form>
END;
}
}

```

13. anyco_ui.inc ファイルを編集します。 ui_print_modify_employee() 関数を追加して、従業員レコードを更新するためのフォームを生成します。

```

function ui_print_modify_employee($empdetails, $posturl)
{
if (!$empdetails) {
echo '<p>No Employee record selected</p>';
}
else {
$fnm = htmlentities($empdetails['FIRST_NAME']);
$lnm = htmlentities($empdetails['LAST_NAME']);
$eml = htmlentities($empdetails['EMAIL']);
$sal = htmlentities($empdetails['SALARY']);
$cpt = htmlentities($empdetails['COMMISSION_PCT']);
$eid = htmlentities($empdetails['EMPLOYEE_ID']);

echo <<<END
<form method="post" action="$posturl">
<table>
<tr>
<td>Employee ID</td>
<td>$eid</td></tr>
<tr>
<td>First Name</td>
<td><input type="text" name="firstname" value="$fnm"></td>
</tr>
<tr>
<td>Last Name</td>
<td><input type="text" name="lastname" value="$lnm"></td>
</tr>
<tr>
<td>Email Address</td>
<td><input type="text" name="email" value="$eml"></td>
</tr>
<tr>

```

```

<td>Salary</td>
<td><input type="text" name="salary" value="$sal"></td>
</tr>
<tr>
<td>Commission (%)</td>
<td><input type="text" name="commpct" value="$cpt"></td>
</tr>
</table>
<input type="hidden" value="{\$empdetails['EMPLOYEE_ID']}"
name="empid">
<input type="submit" value="Save" name="savemodifiedemp">
<input type="submit" value="Cancel" name="cancel">
</form>
END;
}
}

```

14. Anycoアプリケーション・ファイルへの変更を保存し、Webブラウザに次のURLを入力して、変更をテストします。

Windowsの場合:

`http://localhost/chap5/anyco.php`

Linuxの場合:

`http://localhost/~<username>/chap5/anyco.php`

各行にラジオ・ボタンが付いたすべての従業員のリストが表示されます。

Employees					
	Employee ID	Employee Name	Hiredate	Salary	Commission (%)
<input type="radio"/>	100	S. King	17-JUN-87	24,000.00	0
<input type="radio"/>	101	N. Kochhar	21-SEP-89	17,000.00	0
<input type="radio"/>	102	L. De Haan	13-JAN-93	17,000.00	0
<input type="radio"/>	103	A. Hunold	03-JAN-90	9,000.00	0
<input type="radio"/>	104	B. Ernst	21-MAY-91	6,000.00	0
<input type="radio"/>	105	D. Austin	25-JUN-97	4,800.00	0
<input type="radio"/>	106	V. Pataballa	05-FEB-98	4,800.00	0
<input type="radio"/>	107	D. Lorentz	07-FEB-99	4,200.00	0

[chap5_test_emp_001.gifの説明](#)

「Employees」ページの下部にスクロールして、「Modify」、「Delete」、「Insert new employee」の各ボタンを表示します。

	201	M. Hartstein	17-FEB-96	13,000.00	0
	202	P. Fay	17-AUG-97	6,000.00	0
	203	S. Mavris	07-JUN-94	6,500.00	0
	204	H. Baer	07-JUN-94	10,000.00	0
	205	S. Higgins	07-JUN-94	12,000.00	0
	206	W. Gietz	07-JUN-94	8,300.00	0

2005-10-04 13:28:34 Any Co.

[chap5_test_emp_002.gifの説明](#)

15. 新しい従業員レコードを挿入するには、「Insert new employee」をクリックします。

	206	W. Gietz	07-JUN-94	8,300.00	0
--	-----	----------	-----------	----------	---

2005-10-04 13:28:34 Any Co.

[chap5_test_emp_003の説明](#)

従業員レコードを作成または変更する場合は、データベース定義に従って、給与を0（ゼロ）より大きい値にし、コミッションを1より小さい値にする必要があります。コミッションは、小数点以下2桁に められます。「Insert New Employee」ページでは、「Department ID」フィールドは10（デフォルト）、「Hiredate」は現在の日付（デフォルトのデータベース日付書式）、「Salary」は1、「Commission (%)」は0（ゼロ）になっています。次のフィールド値を入力します。

First Name: James

Last Name: Bond

Job: リストから「Programmer」を選択します。

Salary: 1を7000に置き換えます。

「Save」をクリックします。

Insert New Employee

Department ID	<input type="text" value="10"/>
First Name	<input type="text" value="James"/>
Last Name	<input type="text" value="Bond"/>
Hiredate	<input type="text" value="04-OCT-05"/>
Job	<input type="text" value="Programmer"/>
Salary	<input type="text" value="7000"/>
Commission (%)	<input type="text" value="0"/>

2005-10-04 13:31:27

Any Co.

[chap5_test_emp_004.gifの説明](#)

16. 新しい従業員レコードが正常に挿入されると、Webページがリフレッシュされ、フォームにすべての従業員が表示されます。Webページを最後のレコードまでスクロールし、新しい従業員レコードが存在していることを確認します。システムの新しいレコードに割り当てられている従業員IDは、次の例に示す従業員IDとは異なる場合があります。

<input type="radio"/>	206	W. Gietz	07-JUN-94	8,300.00	0
<input type="radio"/>	248	J. Bond	04-OCT-05	7,000.00	0

2005-10-04 13:40:42 Any Co.

[chap5_test_emp_005.gifの説明](#)

17. 新しい従業員レコードを変更するには、そのレコードの があるラジオ・ボタンを選択し、「Modify」をクリックします。

<input type="radio"/>	206	W. Gietz	07-JUN-94	8,300.00	0
<input checked="" type="radio"/>	248	J. Bond	04-OCT-05	7,000.00	0

2005-10-04 13:40:42 Any Co.

[chap5_test_emp_006.gifの説明](#)

18. 「Modify Employee」ページで、「Email Address」フィールドをJBONDに変更し、「Salary」を7100に増やし、「Save」をクリックします。

Modify Employee

Employee ID	248
First Name	James
Last Name	Bond
Email Address	JBOND
Salary	7100
Commission (%)	0

2005-10-04 13:45:04

Any Co.

[chap5_test_emp_007.gifの説明](#)

19. 従業員レコードが正常に更新されると、「Employees」ページが再表示されます。最後の従業員レコードまでスクロールし、James Bondの 与が7,100になっていることを確認します。

<input type="radio"/>	248	J. Bond	04-OCT-05	7,100.00	0
<input type="button" value="Modify"/>	<input type="button" value="Delete"/>	<input type="button" value="Insert new employee"/>			
2005-10-04 13:47:38					Any Co.

[chap5_test_emp_008.gifの説明](#)

20. 新しい従業員レコードを削除するには、そのレコードの にあるラジオ・ボタンを選択し、「Delete」をクリックします。

<input checked="" type="radio"/>	248	J. Bond	04-OCT-05	7,100.00	0
<input type="button" value="Modify"/>	<input type="button" value="Delete"/>	<input type="button" value="Insert new employee"/>			
2005-10-04 13:47:38					Any Co.

[chap5_test_emp_009.gifの説明](#)

削除が正常に完了すると、削除した行は「Employees」ページに再表示された従業員レコードのリストに表示されません。

<input type="radio"/>	206	W. Gietz	07-JUN-94	8,300.00	0
<input type="button" value="Modify"/>	<input type="button" value="Delete"/>	<input type="button" value="Insert new employee"/>			
2005-10-04 13:52:19					Any Co.

[chap5_test_emp_010.gifの説明](#)

「Departments」と「Employees」の結合

この項では、「Employees」と「Departments」の両方のページにアクセスできるようにアプリケーションを変更します。

「Departments」ページと「Employees」ページを結合するには、次の手順を実行します。

1. anyco.phpファイルを編集します。construct_employees()関数の問合せに、department_idを:didというバインド変数の値と比較するWHERE句を含めます。これによって、一度に1つの部門の従業員がページに表示されます。deptidセッション・パラメータ値を取得してバインド変数に移入します。

```
$query =
"SELECT employee_id,
  substr(first_name,1,1) || ' ' || last_name as employee_name,
  hire_date,
  to_char(salary, '9999G999D99') as salary,
  nvl(commission_pct,0) as commission_pct
FROM employees
WHERE department_id = :did
ORDER BY employee_id asc";
```

```
$deptid = $_SESSION['deptid'];
```

2. anyco.phpファイルを編集します。construct_employees()関数で、バインド情報が渡されるようにdb_do_query()関数のコールを更新します。

```
$conn = db_connect();
```

```
$bindargs = array();
array_push($bindargs, array('DID', $deptid, -1));
```

```
$emp = db_do_query($conn, $query, OCI_FETCHSTATEMENT_BY_ROW, $bindargs);
```

3. anyco.phpファイルを編集します。construct_departments()関数で、部門IDをセッション・パラメータに保存します。

```
$_SESSION['currentdept'] = $current;
$_SESSION['deptid'] = $deptid;
```

「Departments」ページの現行の部門IDがセッション・パラメータとして保存され、「Employees」ページで使用されます。

4. anyco.phpファイルを編集します。「Departments」ページおよび「Employees」ページのタイトルに出力する部門名を問い合わせる関数get_dept_name()を作成します。

```
function get_dept_name($conn, $deptid)
{
  $query =
  'SELECT department_name
  FROM departments
  WHERE department_id = :did';

  $conn = db_connect();
  $bindargs = array();
  array_push($bindargs, array('DID', $deptid, -1));
  $dn = db_do_query($conn, $query, OCI_FETCHSTATEMENT_BY_COLUMN, $bindargs);

  return($dn['DEPARTMENT_NAME'][0]);
}
```

5. anyco.phpファイルを編集します。「Employees」ページのヘッダーに部門名が出力されるようにconstruct_employees()関数を変更します。

```
$deptname = get_dept_name($conn, $deptid);
ui_print_header('Employees: '.$deptname);
```

6. anyco.phpファイルを編集します。「Departments」ページのヘッダーに部門名が出力されるようにconstruct_departments()関数を変更します。

エラー・リカバリの追加

エラー管理は、常に設計上の重要な決定事項です。本番システムでは、エラーを分類し、様々な方法で処理する必要がある場合があります。致 的エラーは、標準の「site not available」ページまたはホームページにリダイレクトできます。新しいレコードの作成で発生したデータ・エラーは、無効なフィールドがハイライト表示された該当するフォームに戻すことができます。

ほとんどの本番システムでは、php.iniファイルのdisplay_errors構成オプションをoffに、log_errors構成オプションをonに設定します。

PHP出力バッファリング機能を使用して、関数の実行中にエラー・テキストをトラップできます。ob_start()を使用すると、エラー・テキストが画面に表示されないようにすることができます。エラーが発生した場合は、ob_get_contents()関数を使用して、以前に生成されたエラー・メッセージを後で表示または分析するために文字列に格納できます。

ここで、カスタム・エラー処理関数を使用して新しいページにエラー・メッセージおよびデータベース・エラーを表示するようにアプリケーションを変更します。これによって、エラーがdb*関数から戻され、的に保持されます。

1. anyco_db.incファイルを編集します。出力して終了するのではなく、エラー情報を配列構造に戻すようにdb_error()関数を変更します。ファイルは、次のようになります。

```
function db_error($r = false, $file, $line)
{
    $err = $r ? oci_error($r) : oci_error();

    if (isset($err['message'])) {
        $m = htmlentities($err['message']);
        $c = $err['code'];
    }
    else {
        $m = 'Unknown DB error';
        $c = null;
    }

    $src = array(
        'MESSAGE' => $m,
        'CODE' => $c,
        'FILE' => $file,
        'LINE' => $line
    );
    return $src;
}
```

2. anyco_db.incファイルを編集します。 db_error()関数のすべてのコールに対して、\$eという変数に戻り値を割り当て、各コールの後にreturn false;文を追加します。

```
if (<error test>)
{
    $e = db_error(<handle>, __FILE__, __LINE__);
    return false;
}
```

<error test>パラメータおよび<handle>パラメータは、各コールに現在指定されているものと同じにしてください。__FILE__定数および__LINE__定数を使用すると、開発時に障害の場所を特定するのに役立ちます。この情報は、アプリケーションの本番デプロイメントで致 的エラーが発生した場合に備えてログに記録しておく有効な情報です。

3. anyco_db.incファイルを編集します。すべての関数に\$eパラメータを追加して、エラー情報に戻すことができるようにします。 &参照接頭辞を使用して、結果がコール元関数に戻されるようにします。 各関数の宣言は、次のようになります。

```
function db_connect(&$e) {...}

function db_get_page_data($conn, $sql, $currrownum = 1, $rowsperpage = 1,
    &$e, $bindvars = array()) {...}

function db_do_query($conn, $statement, $resulttype, &$e,
    $bindvars = array()) {...}

function db_execute_statement($conn, $statement, &$e,
    $bindvars = array()) {...}
```

4. anyco_db.incファイルを編集します。 db_get_page_data()関数で、エラー・パラメータ\$eが渡されるようにdb_do_query()関数へのコールを変更します。

```
$r = db_do_query($conn, $query, OCI_FETCHSTATEMENT_BY_ROW, $e, $bindvars);
```

5. anyco_db.incファイルを編集します。すべてのoci_*関数コールに@接頭辞を追加します。次に例を示します。

```
@$r = @oci_execute($stid);
```

@接頭辞を使用すると、戻された結果がそれぞれテストされるため、エラーが表示されなくなります。エラーが表示されないと、誤ったパラメータの使用が されてしまう可能性があるため、この項で行った変更をテストできない場合があります。@接頭辞は追加する必要はありませんが、追加すると、エラーが表示された場合の 後の結果に影響を及 ず可能性があります。

6. anyco.phpファイルを編集します。 エラー情報を処理する関数を作成します。

```
function handle_error($message, $err)
{
    ui_print_header($message);
    ui_print_error($err, $_SERVER['SCRIPT_NAME']);
    ui_print_footer(date('Y-m-d H:i:s'));
}
```

7. anyco.phpファイルを編集します。 追加のエラー・パラメータが含まれるように、 db_*関数へのすべてのコールを変更します。

手順8～15で新しい関数を完成させるため、この手順で行うコード変更はスキップできます。

- すべてのdb_connect()コールをdb_connect(\$err)に変更します。
- すべてのdb_do_query()コールを変更し、4つ目のパラメータとして\$errパラメータを挿入します。たとえば、construct_employees()のコールは次のようになります。

```
$emp = db_do_query($conn, $query,
    OCI_FETCHSTATEMENT_BY_ROW, $err, $bindargs);
```

コールごとに既存のパラメータ値が保持されるように、anyco.phpの他の4つのdb_do_query()コールを変更します。

- すべてのdb_get_page_data()コールを変更し、5つ目のパラメータとして\$errパラメータを挿入します。

```
$dept = db_get_page_data($conn, $query, $current, 1, $err);
```

- すべてのdb_execute_statement()コールを変更し、3つ目のパラメータとして\$errパラメータを挿入します。次に例を示します。

```
$r = db_execute_statement($conn, $statement, $err, $bindargs);
```

8. anyco.phpファイルを編集します。 戻されたエラーが処理されるようにconstruct_departments()関数を変更します。ファイルは、次のようになります。

```
function construct_departments()
{
    if (isset($_SESSION['currentdept']) && isset($_POST['prevdept']) &&
        $_SESSION['currentdept'] > 1)
        $current = $_SESSION['currentdept'] - 1;
    elseif (isset($_SESSION['currentdept']) && isset($_POST['nextdept']))
        $current = $_SESSION['currentdept'] + 1;
    elseif (isset($_POST['showdept']) && isset($_SESSION['currentdept']))
        $current = $_SESSION['currentdept'];
}
```

```

else
    $current = 1;

$query =
"SELECT d.department_id, d.department_name,
       substr(e.first_name,1,1) || ' ' || e.last_name as manager_name,
       c.country_name, count(e2.employee_id) as number_of_employees
FROM departments d, employees e, locations l,
     countries c, employees e2
WHERE d.manager_id = e.employee_id
AND d.location_id = l.location_id
AND d.department_id = e2.department_id
AND l.country_id = c.country_id
GROUP BY d.department_id, d.department_name,
         substr(e.first_name,1,1) || ' ' || e.last_name, c.country_name
ORDER BY d.department_id ASC";

$conn = db_connect($err);

if (!$conn) {
    handle_error('Connection Error', $err);
}
else {
    $dept = db_get_page_data($conn, $query, $current, 1, $err);
    if ($dept === false) {
        // Use === so empty array at end of fetch is not matched
        handle_error('Cannot fetch Departments', $err);
    } else {

        if (!isset($dept[0]['DEPARTMENT_ID']) && $current > 1) {
            // no more records so go back one

            $current--;
            $dept = db_get_page_data($conn, $query, $current, 1, $err);
        }

        $deptid = $dept[0]['DEPARTMENT_ID'];

        $_SESSION['deptid'] = $deptid;
        $_SESSION['currentdept'] = $current;

        $deptname = get_dept_name($conn, $deptid);
        ui_print_header('Department: ' . $deptname);
        ui_print_department($dept[0], $_SERVER['SCRIPT_NAME']);
        ui_print_footer(date('Y-m-d H:i:s'));
    }
}
}
}
}

```

9. anyco.php ファイルを編集します。エラーが処理されるように `construct_employees()` 関数を変更します。ファイルは、次のようになります。

```

function construct_employees()
{
    $query =
"SELECT employee_id,
       substr(first_name,1,1) || ' ' || last_name as employee_name,
       hire_date,
       to_char(salary, '9999G999D99') as salary,
       nvl(commission_pct,0) as commission_pct
FROM employees
WHERE department_id = :did
ORDER BY employee_id asc";

    $deptid = $_SESSION['deptid'];

    $conn = db_connect($err);

    if (!$conn) {
        handle_error('Connection Error', $err);
    }
    else {
        $bindargs = array();
        array_push($bindargs, array('DID', $deptid, -1));
        $emp = db_do_query($conn, $query, OCI_FETCHSTATEMENT_BY_ROW, $err,
            $bindargs);
    }
}

```

```

if (!$emp) {
    handle_error('Cannot fetch Employees', $err);
}
else {
    $deptname = get_dept_name($conn, $deptid);
    ui_print_header('Employees: ' . $deptname);
    ui_print_employees($emp, $_SERVER['SCRIPT_NAME']);
    ui_print_footer(date('Y-m-d H:i:s'));
}
}
}

```

10. anyco.phpファイルを編集します。エラーが処理されるようにconstruct_insert_emp()関数を変更します。ファイルは、次のようになります。

```

function construct_insert_emp()
{
    $deptid = $_SESSION['deptid'];
    $conn = db_connect($err);
    if (!$conn) {
        handle_error('Connection Error', $err);
    }
    else {
        $query = "SELECT job_id, job_title FROM jobs ORDER BY job_title ASC";
        $jobs = db_do_query($conn, $query, OCI_FETCHSTATEMENT_BY_COLUMN, $err);
        $query = "SELECT sysdate FROM dual";
        $date = db_do_query($conn, $query, OCI_FETCHSTATEMENT_BY_COLUMN, $err);

        $emp = array(
            'DEPARTMENT_ID' => $deptid,
            'HIRE_DATE' => $date['SYSDATE'][0],
            'ALLJOBIDS' => $jobs['JOB_ID'],
            'ALLJOBTITLES' => $jobs['JOB_TITLE']
        );

        ui_print_header('Insert New Employee');
        ui_print_insert_employee($emp, $_SERVER['SCRIPT_NAME']);
        ui_print_footer(date('Y-m-d H:i:s'));
    }
}

```

11. anyco.phpファイルを編集します。エラーが処理されるようにinsert_new_emp()関数を変更します。ファイルは、次のようになります。

```

function insert_new_emp()
{
    $statement =
        'INSERT INTO employees
          (employee_id, first_name, last_name, email, hire_date,
           job_id, salary, commission_pct, department_id)
        VALUES (employees_seq.nextval, :fnm, :lnm, :eml, :hdt,
                :jid, :sal, :cpt, :did)';

    $newemp = $_POST;

    $conn = db_connect($err);
    if (!$conn) {
        handle_error('Connect Error', $err);
    }
    else {
        $emailid = $newemp['firstname'] . $newemp['lastname'];

        $bindargs = array();
        array_push($bindargs, array('FNM', $newemp['firstname'], -1));
        array_push($bindargs, array('LNM', $newemp['lastname'], -1));
        array_push($bindargs, array('EML', $emailid, -1));
        array_push($bindargs, array('HDT', $newemp['hiredate'], -1));
        array_push($bindargs, array('JID', $newemp['jobid'], -1));
        array_push($bindargs, array('SAL', $newemp['salary'], -1));
        array_push($bindargs, array('CPT', $newemp['commpct'], -1));
        array_push($bindargs, array('DID', $newemp['deptid'], -1));

        $r = db_execute_statement($conn, $statement, $err, $bindargs);
        if ($r) {

```

```

    construct_employees();
}
else {
    handle_error('Cannot insert employee', $err);
}
}
}

```

12. anyco.php関数を編集します。エラーが処理されるようにconstruct_modify_emp()関数を変更します。ファイルは、次のようになります。

```

function construct_modify_emp()
{
    if (!isset($_POST['emprec'])) { // User did not select a record
        construct_employees();
    }
    else {
        $empid = $_POST['emprec'];

        $query =
            "SELECT employee_id, first_name, last_name, email, hire_date,
              salary, nvl(commission_pct,0) as commission_pct
            FROM employees
            WHERE employee_id = :empid";

        $conn = db_connect($err);
        if (!$conn) {
            handle_error('Connect Error', $err);
        }
        else {
            $bindargs = array();
            array_push($bindargs, array('EMPID', $empid, -1));

            $emp = db_do_query($conn, $query, OCI_FETCHSTATEMENT_BY_ROW, $err,
                $bindargs);

            if (!$emp) {
                handle_error('Cannot find details for employee '.$empid, $err);
            }
            else {
                ui_print_header('Modify Employee ');
                ui_print_modify_employee($emp[0], $_SERVER['SCRIPT_NAME']);
                ui_print_footer(date('Y-m-d H:i:s'));
            }
        }
    }
}

```

13. anyco.phpファイルを編集します。エラーが処理されるようにmodify_emp()関数を変更します。ファイルは、次のようになります。

```

function modify_emp()
{
    $newemp = $_POST;

    $statement =
        "UPDATE employees
        SET first_name = :fnm, last_name = :lnm, email = :eml,
          salary = :sal, commission_pct = :cpt
        WHERE employee_id = :eid";

    $conn = db_connect($err);
    if (!$conn) {
        handle_error('Connect Error', $err);
    }
    else {
        $bindargs = array();
        array_push($bindargs, array('EID', $newemp['empid'], -1));
        array_push($bindargs, array('FNM', $newemp['firstname'], -1));
        array_push($bindargs, array('LNM', $newemp['lastname'], -1));
        array_push($bindargs, array('EML', $newemp['email'], -1));
        array_push($bindargs, array('SAL', $newemp['salary'], -1));
        array_push($bindargs, array('CPT', $newemp['commpct'], -1));

        $r = db_execute_statement($conn, $statement, $err, $bindargs);
    }
}

```

```

if (!$r) {
    handle_error('Cannot update employee '.$newemp['empid'], $err);
}
else {
    construct_employees();
}
}
}

```

14. anyco.phpファイルを編集します。エラーが処理されるようにdelete_emp()関数を変更します。ファイルは、次のようになります。

```

function delete_emp()
{
    if (isset($_POST['emprec'])) { // User did not select a record
        construct_employees();
    }
    else {
        $empid = $_POST['emprec'];

        $conn = db_connect($err);
        if (!$conn) {
            handle_error('Connection Error', $err);
        }
        else {
            $statement = "DELETE FROM employees WHERE employee_id = :empid";
            $bindargs = array();
            array_push($bindargs, array('EMPID', $empid, -1));
            $r = db_execute_statement($conn, $statement, $err, $bindargs);

            if (!$r) {
                handle_error("Error deleting employee $empid", $err);
            }
            else {
                construct_employees();
            }
        }
    }
}

```

15. anyco.phpファイルを編集します。エラーが処理されるようにget_dept_name()関数を変更します。ファイルは、次のようになります。

```

function get_dept_name($conn, $deptid)
{
    $query =
        'SELECT department_name
        FROM departments
        WHERE department_id = :did';

    $conn = db_connect($err);
    if (!$conn) {
        return ('Unknown');
    }
    else {
        $bindargs = array();
        array_push($bindargs, array('DID', $deptid, -1));
        $dn = db_do_query($conn, $query, OCI_FETCHSTATEMENT_BY_COLUMN,
            $err, $bindargs);
        if ($dn == false)
            return ('Unknown');
        else
            return($dn['DEPARTMENT_NAME'][0]);
    }
}

```

16. anyco_ui.incファイルを編集します。新しい関数ui_print_error()を追加します。

```

function ui_print_error($message, $posturl)
{
    if (!$message) {
        echo '<p>Unknown error</p>';
    }
    else {
        echo "<p>Error at line {$message['LINE']} of "

```

```

    .{"$message['FILE']}</p>"; // Uncomment for debugging
    echo "<p>{"$message['MESSAGE']}</p>";
}
echo <<<END
<form method="post" action="$posturl">
<input type="submit" value="Return to Departments" name="showdept">
END;
}

```

END;行の先頭には空白を配置しないでください。END;行の先頭に空白を配置すると、残りのドキュメントが出力対象テキストの一部として処理されます。

- 17. アプリケーション・ファイルへの変更を保存します。ブラウザに次のURLを入力して、変更をテストします。

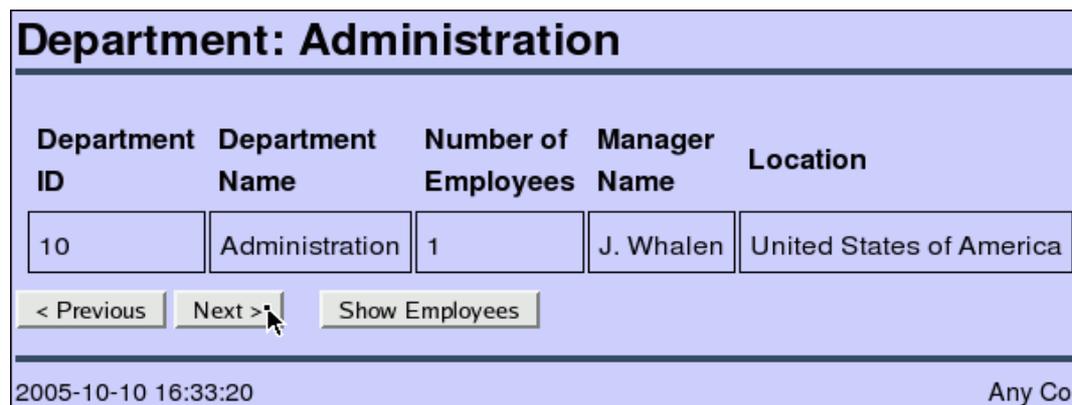
Windowsの場合:

http://localhost/chap5/anyco.php

Linuxの場合:

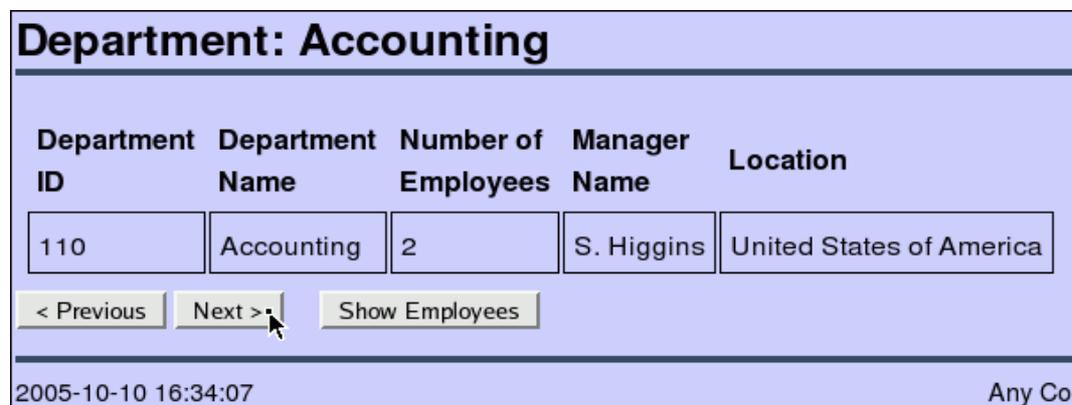
http://localhost/~<username>/chap5/anyco.php

「Departments」ページが表示されます。



[chap5_err_handling_001.gifの説明](#)

- 18. 「Next」をクリックして、最後の部門レコード (IDが110の「Accounting」部門) にナビゲートします。「Next」をクリックして、最後の部門レコードを越えてナビゲートしてみます。



[chap5_err_handling_002.gifの説明](#)

エラー処理によって、最後の部門レコードを越えてナビゲートすることはできません。

- 19. 与0 (ゼロ) の新しい従業員を挿入した場合、または部門IDを存在しないIDに変更した場合は、「Cannot insert employee」というヘッダーの新しいエラー・ページが表示されます。

追加のエラー処理

特定のOracleエラーを個々に処理できます。たとえば、「Employees」ページで「Insert new employee」ボタンをクリックして新しい従業員レコードを作成し、部門IDが存在しない部門に変更した場合は、このエラーをトラップし、よりわかりやすいメッセージを表示できます。

1. anyco.phpファイルを編集します。insert_new_emp()関数のエラー処理を変更します。

```
$r = db_execute_statement($conn, $statement, $err, $bindargs);
if ($r) {
    construct_employees();
}
else {
    if ($err['CODE'] == 2291) { // Foreign key violated
        handle_error("Department {$newemp['deptid']} does not yet exist",
            $err);
    }
    else {
        handle_error('Cannot insert employee', $err);
    }
}
```

2. アプリケーション・ファイルへの変更を保存します。次のURLを入力して、変更をテストします。

Windowsの場合:

<http://localhost/chap5/anyco.php>

Linuxの場合:

<http://localhost/~<username>/chap5/anyco.php>

3. 「Departments」ページで、「Show Employees」をクリックします。

Department ID	Department Name	Number of Employees	Manager Name	Location
10	Administration	1	J. Whalen	United States of America

< Previous Next > Show Employees

2005-10-10 16:37:29 Any Co.

[chap5_err_handling_003.gifの説明](#)

4. 「Employees」ページで、「Insert new employee」をクリックします。

Employees: Administration

Employee ID	Employee Name	Hiredate	Salary	Commission (%)
200	J. Whalen	17-SEP-87	4,400.00	0

2005-10-10 16:37:54 Any Co.

[chap5_err_handling_004.gifの説明](#)

5. 「Insert New Employee」 ページで、次に示すように従業員詳細を入力し、「Department ID」を99に設定して、「Save」をクリックします。

Insert New Employee

Department ID	99
First Name	New
Last Name	Person
Hiredate	10-OCT-05
Job	Accountant
Salary	1000
Commission (%)	0

2005-10-10 16:38:06 Any Co.

[chap5_err_handling_005.gifの説明](#)

次のエラー・ページが表示されます。

Department 99 does not yet exist

Error at line 86 of /home/gstokol/public_html/chap5/anyco_db.inc

ORA-02291: integrity constraint (HR.EMP_DEPT_FK) violated - parent key not found

2005-10-10 16:39:15 Any Co.

[chap5_err_handling_006.gifの説明](#)

「Return to Departments」をクリックして「Departments」ページに戻り、「Show Employees」をクリックして新しい従業員レコードが「Administration」部門に追加されていないことを確認できます。

6 ストアド・プロシージャおよびストアド・ファンクションの実行

この章では、PHPおよびOracle Databaseを使用してストアド・プロシージャおよびストアド・ファンクションを実行する方法について説明します。内容は次のとおりです。

- [PL/SQLを使用したビジネス・ロジックの取得](#)
- [PL/SQL参照カーソルを使用した結果セットの戻し](#)

各従業員の報酬を計算するPL/SQLファンクションおよび従業員レコードの参照カーソルを戻すPL/SQLプロシージャでAnycoアプリケーションを拡張します。

PL/SQLを使用したビジネス・ロジックの取得

Oracle PL/SQLプロシージャおよびファンクションを使用すると、すべてのクライアント・プログラムで使用できるビジネス・ロジックをデータベースに保存できます。これによって、データベースとPHP間で転送するデータ量を減らし、パフォーマンスを向上させることができます。

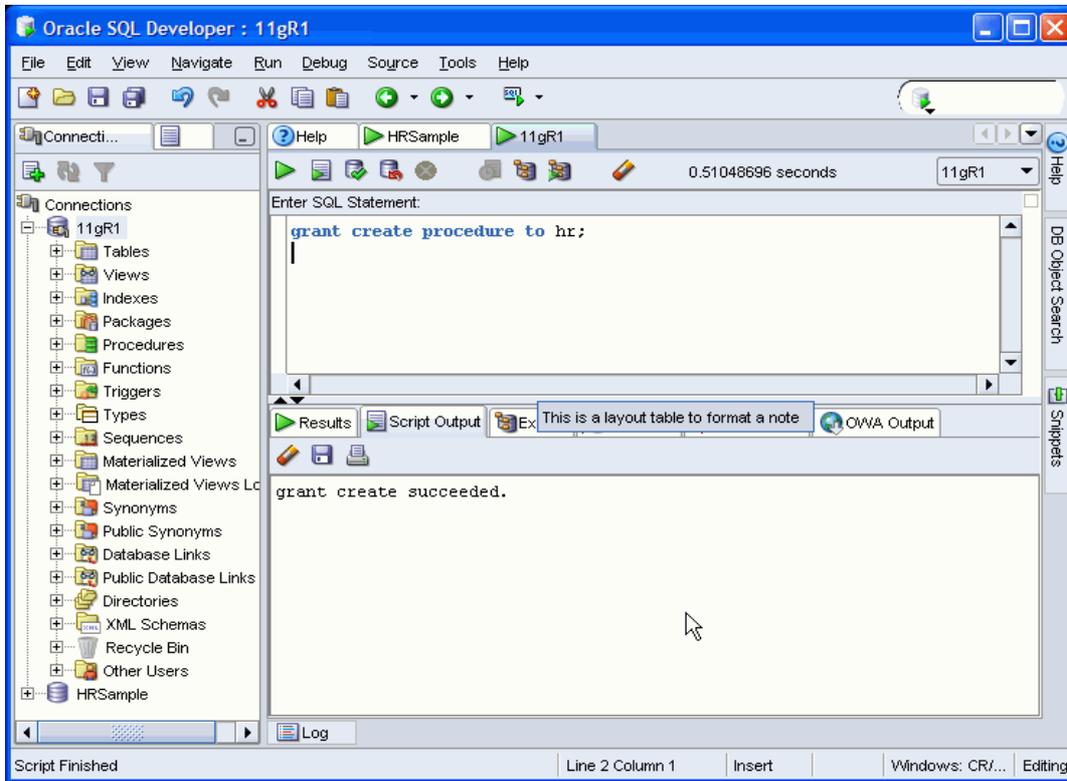
この項では、各従業員の合計報酬を計算して表示するPL/SQLストアド・ファンクションを作成します。

各従業員の合計報酬を表示するには、次の手順を実行します。

PHPアプリケーションは、HRユーザーとしてデータベースに接続します。DBA権限を持つユーザーとして、HRアカウントをロック解除する必要があります場合があります。HRユーザーをロック解除するには、次の手順を実行します。

1. SQL Developerを起動し、Oracle Databaseへの接続をオープンします。
2. **system**ユーザーとしてOracle Databaseにログインします。
3. SQLワークシートまたはSQL*Plusを開き、次のgrant文を実行してHRユーザーにcreate procedure権限を割り当てます。

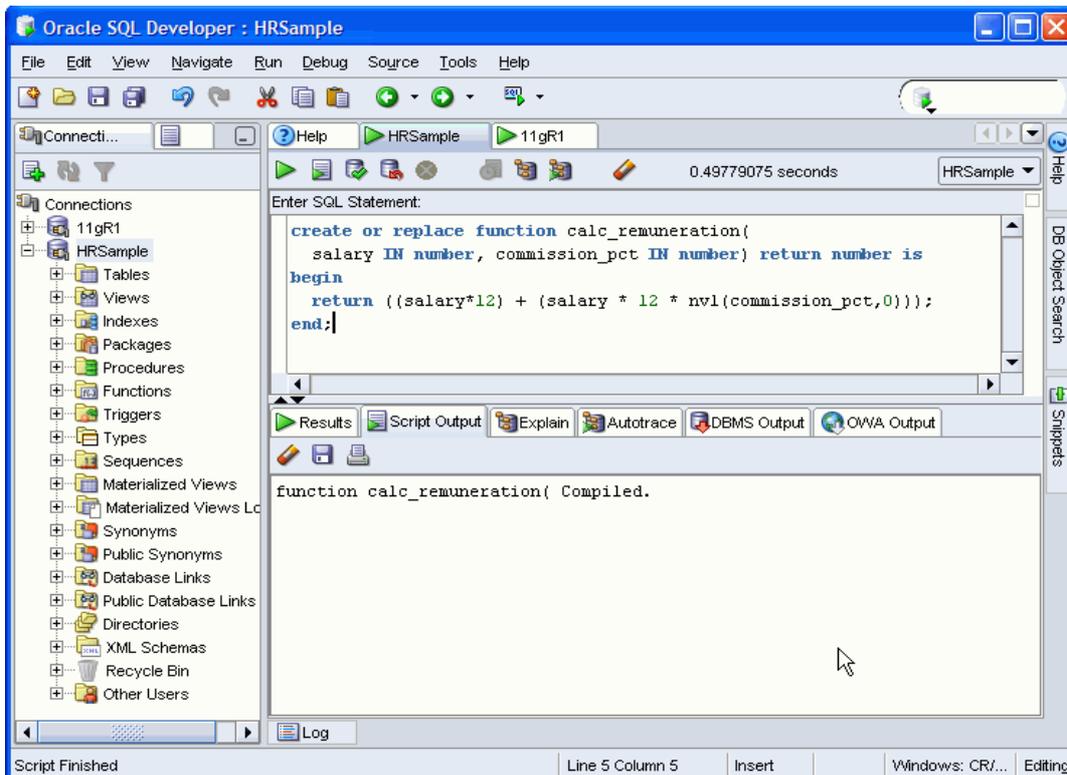
```
grant create procedure to hr;
```



[chap6_hrgrantproc.gifの説明](#)

4. hrユーザーとしてHRサンプル・スキーマにログインします。
5. SQLワークシートまたはSQL*Plusを開き、次のテキストを入力してcalc_remuneration()関数を作成します。

```
create or replace function calc_remuneration(
  salary IN number, commission_pct IN number) return number is
begin
  return ((salary*12) + (salary * 12 * nvl(commission_pct,0)));
end;
```



[chap6_hrcalcrem.gifの説明](#)

6. chap6ディレクトリを作成し、chap5からアプリケーション・ファイルをコピーし、新しく作成したディレクトリに移動します。

Windowsの場合:

```
mkdir c:\program files\Apache Group\Apache2\htdocs\chap6
cd c:\program files\Apache Group\Apache2\htdocs\chap6
copy ..\chap5\* .
```

Linuxの場合:

```
mkdir $HOME/public_html/chap6
cd $HOME/public_html/chap6
cp ../chap5/* .
```

7. anyco.phpファイルを編集します。行が戻されるたびにPL/SQLファンクションがコールされるようにconstruct_employees()関数の問合せを変更します。

```
$query =
"SELECT employee_id,
      substr(first_name,1,1) || ' ' || last_name as employee_name,
      hire_date,
      to_char(salary, '9999G999D99') as salary,
      nvl(commission_pct,0) as commission_pct,
      to_char(calc_remuneration(salary, commission_pct),'9999G999D99')
      as remuneration
FROM employees
WHERE department_id = :did
ORDER BY employee_id ASC";
```

8. anyco_ui.incファイルを編集します。ui_print_employees()関数で、表に「Remuneration」列を追加し、従業員ごとに「remuneration」フィールドが表示されるようにforeachループを変更します。

```
echo <<<END
<form method="post" action="$posturl">
<table>
<tr>
<th>&nbsp;&nbsp;&nbsp;</th>
<th>Employee<br>ID</th>
<th>Employee<br>Name</th>
<th>Hiredate</th>
<th>Salary</th>
<th>Commission<br>(%)</th>
<th>Remuneration</th>
</tr>
END;

// Write one row per employee
foreach ($employeeerecords as $emp) {
echo '<tr>';
echo '<td><input type="radio" name="emprec"
      value="'.htmlentities($emp['EMPLOYEE_ID']).'"></td>';
echo '<td align="right">'.htmlentities($emp['EMPLOYEE_ID']).'</td>';
echo '<td>'.htmlentities($emp['EMPLOYEE_NAME']).'</td>';
echo '<td>'.htmlentities($emp['HIRE_DATE']).'</td>';
echo '<td align="right">'.htmlentities($emp['SALARY']).'</td>';
echo '<td align="right">'.htmlentities($emp['COMMISSION_PCT']).'</td>';
echo '<td align="right">'.htmlentities($emp['REMUNERATION']).'</td>';
echo '</tr>';
}
```

9. アプリケーション・ファイルへの変更を保存します。ブラウザで、次のURLを入力して、アプリケーションをテストします。

Windowsの場合:

<http://localhost/chap6/anyco.php>

Linuxの場合:

http://localhost/~<username>/chap6/anyco.php

10. 「Departments」 ページで、「Show Employees」をクリックします。

Department: Administration				
Department ID	Department Name	Number of Employees	Manager Name	Location
10	Administration	1	J. Whalen	United States of America

< Previous Next > Show Employees

2005-10-10 22:12:54 Any Co.

[chap6_stored_proc_test_001.gifの説明](#)

「Employees」 ページで、従業員の報酬が最後の列に表示されます。

Employees: Administration					
Employee ID	Employee Name	Hiredate	Salary	Commission (%)	Remuneration
 200	J. Whalen	17-SEP-87	4,400.00	0	52,800.00

Modify Delete Insert new employee Return to Departments

2005-10-10 22:14:31 Any Co.

[chap6_stored_proc_test_002.gifの説明](#)

PL/SQL参照カーソルを使用した結果セットの戻し

PL/SQLブロックから参照カーソルとして問合せデータを戻し、PHPに表示できます。この操作は、データセットで複雑な機能が必要な場合または複数のアプリケーション・プログラムで同じ問合せを使用する場合に有効です。

PL/SQLの参照カーソルは、カーソル変数に割り当てられているタイプ定義です。パッケージ仕様部内にPL/SQLタイプを宣言して、パッケージ本体などの他のPL/SQL構造で再利用することがよく行われます。

この項では、参照カーソルを使用して、特定の部門の従業員を取り出します。

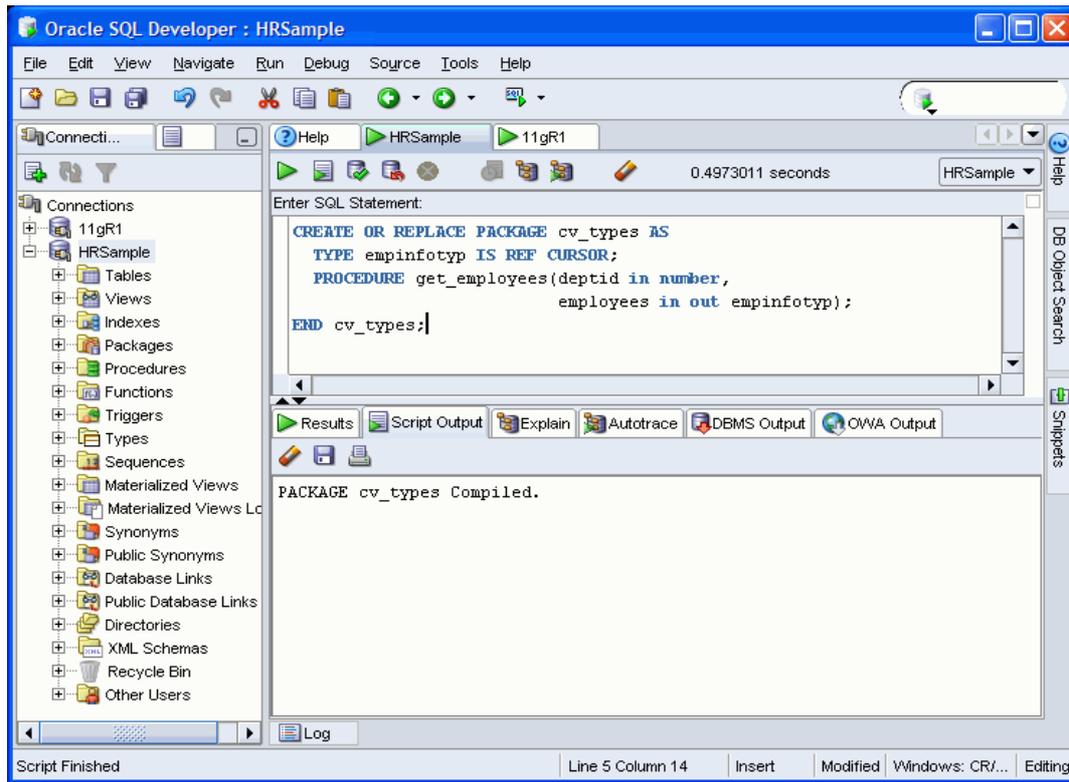
PL/SQLパッケージ仕様部およびパッケージ本体を作成し、参照カーソルを使用して特定の部門の従業員を取り出すには、次の手順を実行します。

1. SQL Developerを起動し、hrユーザーとしてHRサンプル・スキーマにログインします。
2. SQLワークシートまたはSQL*Plusを開き、次のテキストを入力してPL/SQLパッケージcv_typesを作成します。

```
CREATE OR REPLACE PACKAGE cv_types AS
TYPE empinfotyp IS REF CURSOR;
PROCEDURE get_employees(deptid in number,
employees in out empinfotyp);
```

END cv_types;

「Run」をクリックします。

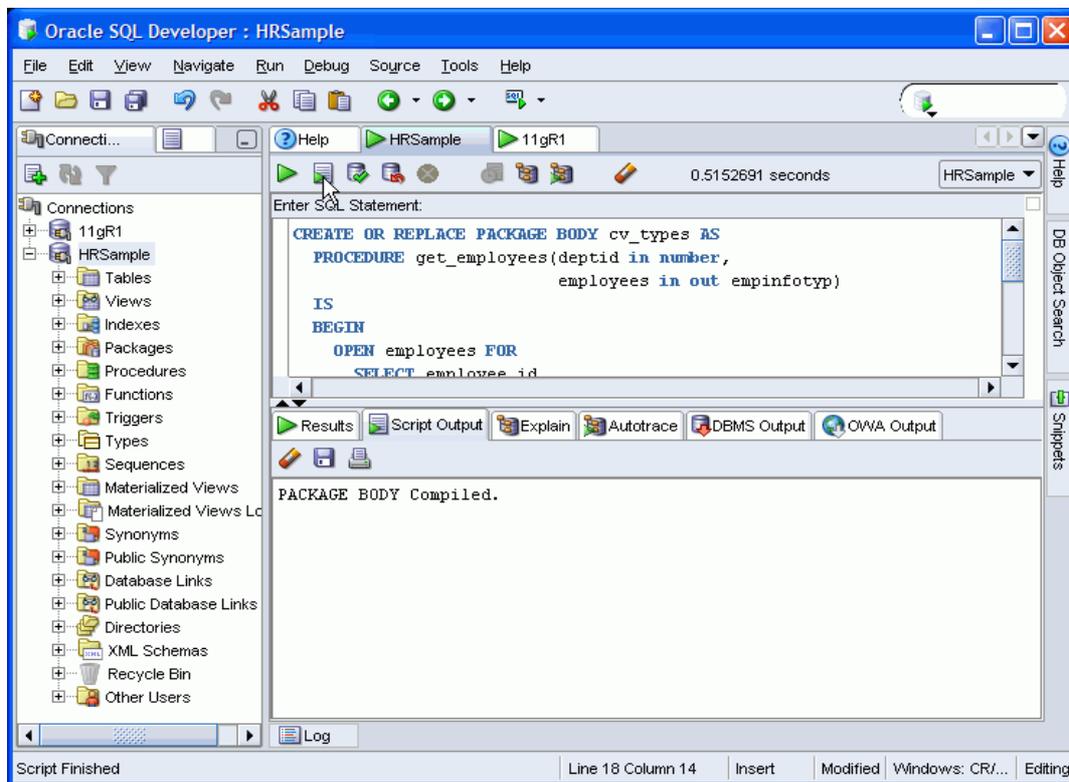


[chap6_hrcreatepack.gifの説明](#)

3. SQLワークシートで、次のテキストを入力してPL/SQLパッケージ本体cv_typesを作成します。

```
CREATE OR REPLACE PACKAGE BODY cv_types AS
  PROCEDURE get_employees(deptid in number,
    employees in out empinfotyp)
  IS
  BEGIN
  OPEN employees FOR
  SELECT employee_id,
    substr(first_name,1,1) || '.' || last_name as employee_name,
    hire_date,
    to_char(salary, '999G999D99') as salary,
    NVL(commission_pct,0) as commission_pct,
    to_char(calc_remuneration(salary, commission_pct),
      '9999G999D99') as remuneration
  FROM employees
  WHERE department_id = deptid
  ORDER BY employee_id ASC;
  END get_employees;
END cv_types;
```

「Run」をクリックします。



chap6_hrcreatebody.gifの説明

4. anyco_db.incファイルを編集します。 PL/SQLパッケージ・プロシージャをコールする新しいPHP関数を作成します。

```

// Use ref cursor to fetch employee records
// All records are retrieved - there is no paging in this example
function db_get_employees_rc($conn, $deptid, &$e)
{
  // Execute the call to the stored procedure
  $stmt = "BEGIN cv_types.get_employees($deptid, :rc); END;";
  $stid = @oci_parse($conn, $stmt);
  if (!$stid) {
    $e = db_error($conn, __FILE__, __LINE__);
    return false;
  }
  $refcur = oci_new_cursor($conn);
  if (!$stid) {
    $e = db_error($conn, __FILE__, __LINE__);
    return false;
  }
  $r = @oci_bind_by_name($stid, 'RC', $refcur, -1, OCI_B_CURSOR);
  if (!$r) {
    $e = db_error($stid, __FILE__, __LINE__);
    return false;
  }
  $r = @oci_execute($stid);
  if (!$r) {
    $e = db_error($stid, __FILE__, __LINE__);
    return false;
  }
  // Now treat the ref cursor as a statement resource
  $r = @oci_execute($refcur, OCI_DEFAULT);
  if (!$r) {
    $e = db_error($refcur, __FILE__, __LINE__);
    return false;
  }
  $r = @oci_fetch_all($refcur, $employeerecords, null, null,
    OCI_FETCHSTATEMENT_BY_ROW);
  if (!$r) {
    $e = db_error($refcur, __FILE__, __LINE__);
    return false;
  }
  return ($employeerecords);
}

```

db_get_employees_rc()関数は、次の無名（名前が指定されていない）PL/SQLブロックを実行します。

```
BEGIN cv_types.get_employees($deptid, :rc); END;
```

BEGIN ENDブロック内のPL/SQL文は、ストアPL/SQLパッケージ・プロシージャcv_types.get_employees()をコールします。これによって、PHP変数\$refcurにOCI_B_CURSOR参照カーソル・バインド変数が戻されます。

\$refcur変数は、oci_parse()によって戻される文ハンドルと同様に処理されます。PHPでSQL問合せを実行した場合と同様に、実行操作およびフェッチ操作に使用されます。

- anyco.phpファイルを編集します。construct_employees()関数で、問合せテキストおよびバインド引数を削除します。ファイルは、次のようになります。

```
function construct_employees()
{
    $deptid = $_SESSION['deptid'];
    $conn = db_connect($err);
    if (!$conn) {
        handle_error('Connection Error', $err);
    }
    else {
        $emp = db_get_employees_rc($conn, $deptid, $err);

        if (!$emp) {
            handle_error('Cannot fetch Employees', $err);
        }
        else {
            $deptname = get_dept_name($conn, $deptid);

            ui_print_header('Employees: ' . $deptname);
            ui_print_employees($emp, $_SERVER['SCRIPT_NAME']);
            ui_print_footer(date('Y-m-d H:i:s'));
        }
    }
}
```

- アプリケーション・ファイルへの変更を保存します。ブラウザで、次のURLを入力して、アプリケーションをテストします。

Windowsの場合:

<http://localhost/chap6/anyco.php>

Linuxの場合:

<http://localhost/~<username>/chap6/anyco.php>

- 「Departments」ページで、「Next」をクリックして「Marketing」部門ページにナビゲートします。

Department ID	Department Name	Number of Employees	Manager Name	Location
10	Administration	1	J. Whalen	United States of America

< Previous Next > Show Employees

2005-10-10 23:27:47 Any Co.

[chap6_refcursor_005.gifの説明](#)

- 「Marketing」部門ページで、「Show Employees」をクリックします。

Department: Marketing

Department ID	Department Name	Number of Employees	Manager Name	Location
20	Marketing	2	M. Hartstein	Canada

2005-10-10 23:28:36

Any Co.

[chap6_refcursor_006.gifの説明](#)

「Employees」 ページに、従業員のページが以前と同様に表示されます。

Employees: Marketing

Employee ID	Employee Name	Hiredate	Salary	Commission (%)	Remuneration
 201	M. Hartstein	17-FEB-96	13,000.00	0	156,000.00
 202	P. Fay	17-AUG-97	6,000.00	0	72,000.00

2005-10-10 23:30:32

Any Co.

[chap6_refcursor_007.gifの説明](#)

7 イメージのロード

この章では、新しい従業員レコードのJPEGイメージをアップロードして「Employees」ページに表示するためにアプリケーションを変更する方法について説明します。内容は次のとおりです。

- [BLOBを使用した従業員イメージの保存およびロード](#)
- [イメージのサイズ変更](#)

BLOBを使用した イメージの およびロード

この項では、従業員のレコードに写真を格納できるようにアプリケーション・コードを変更します。

従業員レコードに従業員のイメージを格納できるようにするには、次の手順を実行します。

1. chap7ディレクトリを作成し、chap6からアプリケーション・ファイルをコピーし、新しく作成したディレクトリに移動します。

Windowsの場合:

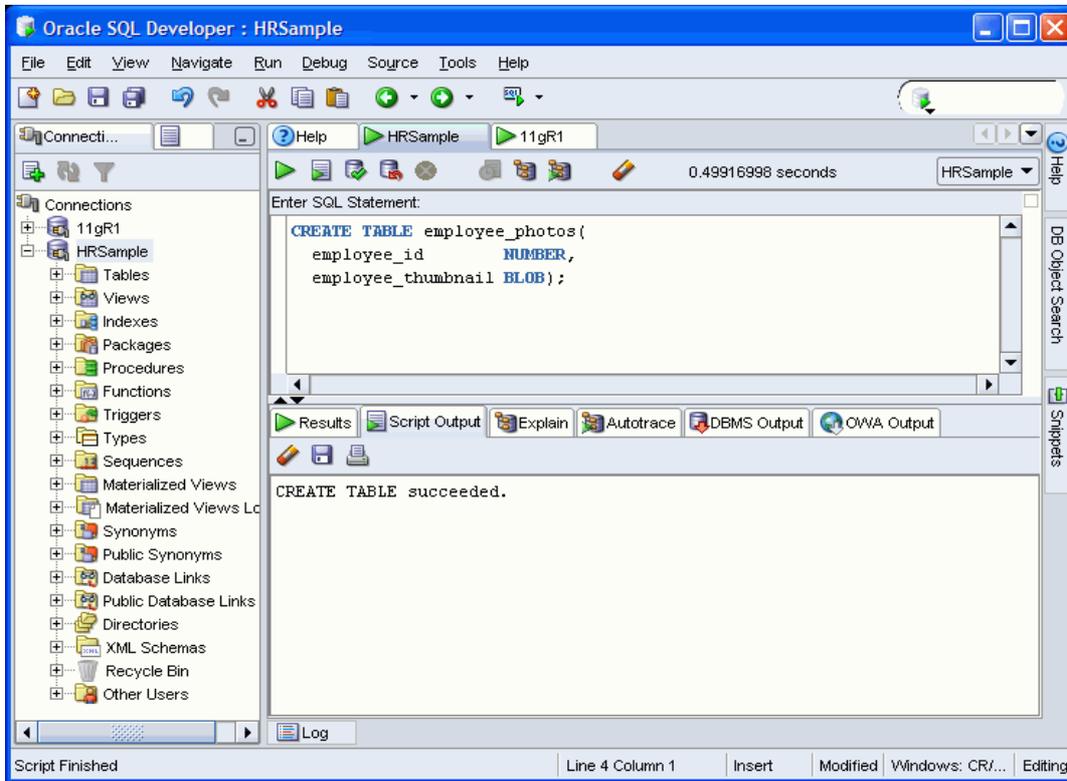
```
mkdir c:\program files\Apache Group\Apache2\htdocs\chap7
cd c:\program files\Apache Group\Apache2\htdocs\chap7
copy ..\chap6\* .
```

Linuxの場合:

```
mkdir $HOME/public_html/chap7
cd $HOME/public_html/chap7
cp ../chap6/* .
```

2. SQL Developerを起動し、HRサンプル・スキーマへの接続をオープンします。
3. **hr**ユーザーとしてHRサンプル・スキーマにログインします。
4. SQLワークシートを開き、次のCREATE TABLE文を入力して、従業員イメージを格納するための新しい表を作成します。

```
CREATE TABLE employee_photos(
employee_id NUMBER,
employee_thumbnail BLOB);
```



chap7_hrcreatetable.gifの説明

- HRユーザーがこのコマンドを実行するには、CREATE TABLE権限が必要です。「insufficient privileges」エラー・メッセージが戻された場合は、HRユーザーとしてログアウトしてから、systemとしてログインし、次のGRANTコマンドを実行します。

```
GRANT create table TO hr;
```

次に、再度HRとしてログインして、CREATE TABLE文を実行します。

- anyco_ui.incファイルを編集します。 ui_print_employees()関数のEMPLOYEES表に「Photograph」列を追加します。

```
<th>Commission<br>(%)</th>
<th>Remuneration</th>
<th>Photograph</th>
```

「Photograph」列のデータは、タグ付きで移入されます。このタグでは、src属性が新しいanyco_im.phpファイルへのURL参照として定義されているため、従業員レコードごとにイメージが表示されます。

- anyco_ui.incファイルを編集します。 ui_print_employees()関数に、従業員IDをパラメータとして指定して、anyco_im.phpファイルを参照するタグを生成するコードを追加します。

```
echo '<td align="right">'
    .htmlentities($emp['REMUNERATION']).'</td>';
echo '<td></td>';
```

- anyco_ui.incファイルを編集します。新しい従業員レコードの作成時にイメージをアップロードできるようにするには、ui_print_insert_employee()関数の<form>タグにenctype属性を追加します。

```
<form method="post" action="$posturl" enctype="multipart/form-data">
```

フォームの下部にアップロード用のフィールドを追加し、その入力タイプをfileに設定します。

```
<tr>
<td>Commission (%)</td>
<td><input type="text" name="compct" value="0" size="20"></td>
```

```

</tr>
<tr>
<td>Photo</td>
<td><input type="file" name="empphoto"></td>
</tr>

```

9. anyco_im.php fileファイルを作成します。このファイルは、URLパラメータとして従業員IDをとり、その従業員レコードの「Photograph」列からイメージを読み取り、表示するサムネイル・イメージを戻します。

```

<?php // anyco_im.php

require('anyco_cn.inc');
require('anyco_db.inc');
construct_image();

function construct_image()
{
    if (!isset($_GET['showempphoto'])) {
        return;
    }

    $empid = $_GET['showempphoto'];

    $conn = db_connect($err);

    if (!$conn) {
        return;
    }

    $query =
        'SELECT employee_thumbnail
        FROM employee_photos
        WHERE employee_id = :eid';

    $stid = oci_parse($conn, $query);
    $r = oci_bind_by_name($stid, ":eid", $empid, -1);
    if (!$r) {
        return;
    }
    $r = oci_execute($stid, OCI_DEFAULT);
    if (!$r) {
        return;
    }

    $arr = oci_fetch_row($stid);
    if (!$arr) {
        return; // photo not found
    }

    $result = $arr[0]->load();

    // If any text (or whitespace!) is printed before this header is sent,
    // the text is not displayed. The image also is not displayed properly.
    // Comment out the "header" line to see the text and debug.
    header("Content-type: image/JPEG");
    echo $result;
}

?>

```

construct_image()関数は、OCI-Lob->load()関数を使用して、イメージ・データであるOracle LOBデータを取り出します。PHPのheader()関数は、HTTPレスポンス・ヘッダーにMIMEタイプを設定して、ブラウザがデータをJPEGイメージとして認識できるようにします。

他のイメージ・タイプを表示する場合は、そのタイプに応じてContent-typeを変更する必要があります。

10. anyco_db.incファイルを編集します。EMPLOYEE_PHOTOS表にイメージを挿入する新しい関数db_insert_thumbnail()を追加します。

```

function db_insert_thumbnail($conn, $empid, $imgfile, &$e)
{
    $lob = oci_new_descriptor($conn, OCI_D_LOB);
    if (!$lob) {
        $e = db_error($conn, __FILE__, __LINE__);
    }
}

```

```

return false;
}

$insstmt =
'INSERT INTO employee_photos (employee_id, employee_thumbnail)
VALUES(:eid, empty_blob())
RETURNING employee_thumbnail into :etn';

$stmt = oci_parse($conn, $insstmt);
$r = oci_bind_by_name($stmt, ':etn', $lob, -1, OCI_B_BLOB);
if (!$r) {
    $e = db_error($stid, __FILE__, __LINE__);
    return false;
}
$r = oci_bind_by_name($stmt, ':eid', $empid, -1);
if (!$r) {
    $e = db_error($stid, __FILE__, __LINE__);
    return false;
}
$r = oci_execute($stmt, OCI_DEFAULT);
if (!$r) {
    $e = db_error($stid, __FILE__, __LINE__);
    return false;
}

if (!$lob->savefile($imgfile)) {
    $e = db_error($stid, __FILE__, __LINE__);
    return false;
}
$lob->free();

return true;
}

```

新しいEMPLOYEE_PHOTOS表とEMPLOYEES表を結び付けるには、両方の表で同じ従業員IDを使用する必要があります。

11. anyco_db.incファイルを編集します。OUTバインド変数値がデータベースから戻されるように、db_execute_statement()関数の\$bindvarsパラメータを&\$bindvarsに変更します。この関数の下部に、戻されたバインド変数値を設定するループを追加します。

```

function db_execute_statement($conn, $statement, &$e, &$bindvars = array())
{
    ...
    $r = @oci_execute($stid);
    if (!$r) {
        $e = db_error($stid, __FILE__, __LINE__);
        return false;
    }
    $outbinds = array();
    foreach ($bindvars as $b) {
        $outbinds[$b[0]] = $$b[0];
    }
    $bindvars = $outbinds;
    return true;
}

```

12. anyco.phpファイルを編集します。バインド変数:neweidに新しい従業員IDが戻されるように、insert_new_emp()関数のINSERT文を変更します。この値がイメージとともに新しいEMPLOYEE_PHOTOS表に挿入されます。

```

$statement =
'INSERT INTO employees
(employee_id, first_name, last_name, email, hire_date,
job_id, salary, commission_pct, department_id)
VALUES (employees_seq.nextval, :fnm, :lnm, :eml, :hdt,
:jid, :sal, :cpt, :did)
RETURNING employee_id into :neweid';

```

また、insert_new_emp()関数で、array_push()関数へのコールを追加して、新しいバインド変数NEWWEIDをarray_push()コールのリストの最後に設定します。

```

array_push($bindargs, array('CPT', $newemp['commpct'], -1));
array_push($bindargs, array('DID', $newemp['deptid'], -1));
array_push($bindargs, array('NEWWEID', null, 10));

```

NEWIDの値はINSERT文のRETURNING句で取り出されるため、その初期値はNULLに設定します。長さは、戻り値に十分な桁を確保するために10に設定します。

13. anyco.phpファイルを編集します。insert_new_emp()関数で、db_execute_statement()コールとconstruct_employees()コールの間にサムネイル・イメージを挿入するコールを追加します。

```
$r = db_execute_statement($conn, $statement, $err, $bindargs);  
if ($r) {  
    $r = db_insert_thumbnail($conn, $bindargs['NEWID'],  
        $_FILES['empphoto']['tmp_name'], $e);  
    construct_employees();  
}
```

14. ブラウザに、次のアプリケーションURLを入力します。

Windowsの場合:

<http://localhost/chap7/anyco.php>

Linuxの場合:

<http://localhost/~<username>/chap7/anyco.php>

15. 「Departments」 ページで、「Show Employees」をクリックして「Employees」 ページにナビゲートします。

Department ID	Department Name	Number of Employees	Manager Name	Location
10	Administration	1	J. Whalen	United States of America

< Previous Next > Show Employees

2005-10-11 11:18:29 Any Co.

[chap7_loading_005.gifの説明](#)

16. 「Employees」 ページで、新しい従業員レコードを挿入するには、「Insert new employee」をクリックします。

Employee ID	Employee Name	Hiredate	Salary	Commission (%)	Remuneration	Photograph
200	J. Whalen	17-SEP-87	4,400.00	0	52,800.00	Employee photo

Modify Delete Insert new employee Return to Departments

2005-10-11 11:19:31 Any Co.

[chap7_loading_006.gifの説明](#)

17. 「Insert New Employee」 フォームでは、データベースにアップロードするシステム上のサムネイル・イメージを選択できます。次の各フィールドに独自の値を入力するか、または表示されている値を使用します。「Browse」をクリックします。

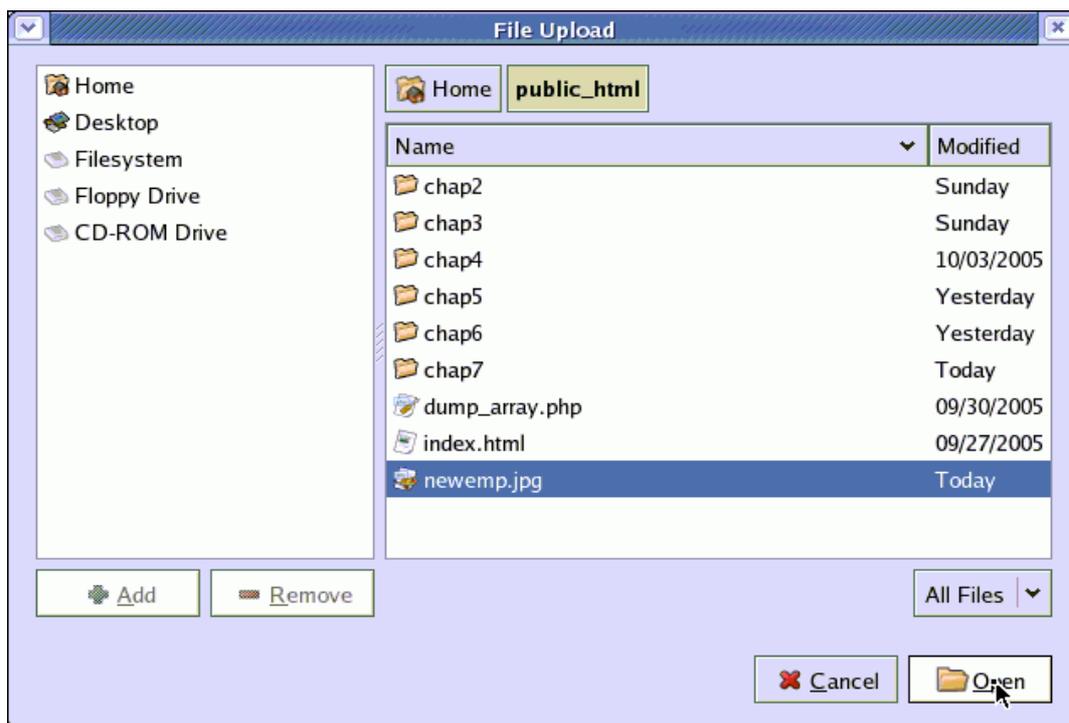
Insert New Employee

Department ID	<input type="text" value="10"/>
First Name	<input type="text" value="Glenn"/>
Last Name	<input type="text" value="Stokol"/>
Hiredate	<input type="text" value="11-OCT-05"/>
Job	<input type="text" value="Programmer"/>
Salary	<input type="text" value="8000"/>
Commission (%)	<input type="text" value="0"/>
Photo	<input type="text"/> <input type="button" value="Browse.."/>

2005-10-11 11:21:05 Any Co.

[chap7_loading_007.gifの説明](#)

18. 「File Upload」 ウィンドウで、JPEGイメージ・ファイルを参照して選択し、「Open」をクリックします。



[chap7_loading_008.gifの説明](#)

19. 「Insert New Employee」 ページで、「Save」をクリックします。

Insert New Employee

Department ID	<input type="text" value="10"/>
First Name	<input type="text" value="Chris"/>
Last Name	<input type="text" value="Jones"/>
Hiredate	<input type="text" value="11-OCT-05"/>
Job	<input type="text" value="Marketing Manager"/>
Salary	<input type="text" value="9000"/>
Commission (%)	<input type="text" value="0"/>
Photo	<input type="text" value="/home/gstokol/public_html"/> <input type="button" value="Browse..."/>

2005-10-11 12:32:04 Any Co.

[chap7_loading_022.gifの説明](#)

「Employees」 ページに、元のサイズのイメージを含む新しい従業員レコードが表示されます。

Employees: Administration

	Employee ID	Employee Name	Hiredate	Salary	Commission (%)	Remuneration	Photograph
	200	J. Whalen	17-SEP-87	4,400.00	0	52,800.00	Employee photo
	209	G. Stokol	11-OCT-05	8,000.00	0	96,000.00	

2005-10-11 12:27:16 Any Co.

[chap7_loading_009.gifの説明](#)

イメージのサイ 変更

この項では、アプリケーション・コードをさらに変更して、指定したイメージからサムネイル・イメージを作成し、従業員のレコードにそのサムネイル・イメージを格納します。

PHPのGDグラフィックス拡張モジュールを使用して、従業員イメージをサイズ変更します。

1. Apacheを再起動します。ApacheMonitorユーティリティを使用するか、またはWindowsのサービスを使用できます。

ApacheMonitorユーティリティを使用するには、Apacheのbinディレクトリに移動し、ApacheMonitor.exeをダブルクリックします。デフォルトのインストールでは、Apacheのbinディレクトリはc:\Program Files\Apache Group\Apache2\binです。

Windowsのサービスにアクセスするには、Windowsの「スタート」メニューで、「スタート」→「コントロールパネル」→「管理ツール」→「サービス」を選択します。「標準」タブを選択します。「Apache2 HTTP Server」を右クリックし、「再起動」を選択します。

2. anyco_db.incファイルを編集します。イメージをサイズ変更してサムネイル・イメージを作成するには、db_insert_thumbnail()関数で\$lob->savefile(\$imgfile)へのコールの前に次のコードを追加します。

```
$r = oci_execute($stmt, OCI_DEFAULT);
if (!$r) {
    $e = db_error($stid, __FILE__, __LINE__);
    return false;
}

// Resize the image to a thumbnail
define('MAX_THUMBNAI_DIMENSION', 100);
$src_img = imagecreatefromjpeg($imgfile);
list($w, $h) = getimagesize($imgfile);
if ($w > MAX_THUMBNAI_DIMENSION || $h > MAX_THUMBNAI_DIMENSION)
{
    $scale = MAX_THUMBNAI_DIMENSION / (($h > $w) ? $h : $w);
    $nw = $w * $scale;
    $nh = $h * $scale;

    $dest_img = imagecreatetruecolor($nw, $nh);
    imagecopyresampled($dest_img, $src_img, 0, 0, 0, 0, $nw, $nh, $w, $h);

    imagejpeg($dest_img, $imgfile); // overwrite file with new thumbnail

    imagedestroy($src_img);
    imagedestroy($dest_img);
}

if (!$lob->savefile($imgfile)) {
    ...
}
```

imagecreatefromjpeg()関数を使用して、JPEGファイルを読み取り、後続のGD関数で使用する内部表現を作成しています。次に、最長側面が100ピクセル以下の新しい次元を計算しています。imagecreatetruecolor()関数を使用して、その新しいサイズのテンプレート・イメージを作成しています。imagecopyresampled()関数を使用して元のイメージのデータをそのテンプレート・イメージにサンプリングして、サムネイル・イメージを作成しています。サムネイル・イメージを元のファイルに書き戻し、イメージの内部表現を解放しています。

db_insert_thumbnail()関数の既存のコードは、以前の実装と同様に、データベースにイメージ・ファイルをアップロードします。

3. ブラウザに次のURLを入力して、アプリケーションの変更をテストします。

Windowsの場合:

<http://localhost/chap7/anyco.php>

Linuxの場合:

<http://localhost/~<username>/chap7/anyco.php>

4. 「Departments」 ページで、「Show Employees」をクリックして、「Employees」 ページにナビゲートします。

Department: Administration

Department ID	Department Name	Number of Employees	Manager Name	Location
10	Administration	2	J. Whalen	United States of America

2005-10-11 12:29:43 Any Co.

[chap7_loading_018.gifの説明](#)

5. 「Employees」 ページで、新しい従業員レコードを挿入するには、「Insert new employee」をクリックします。

Employees: Administration

Employee ID	Employee Name	Hiredate	Salary	Commission (%)	Remuneration
 200	J. Whalen	17-SEP-87	4,400.00	0	52,800.00
 209	G. Stokol	11-OCT-05	8,000.00	0	96,000.00

2005-10-11 12:30:46 Any Co.

[chap7_loading_019.gifの説明](#)

6. 新しく従業員の詳細を入力するか、または表示されている値を使用します。従業員イメージを参照するには、「Browse」をクリックします。

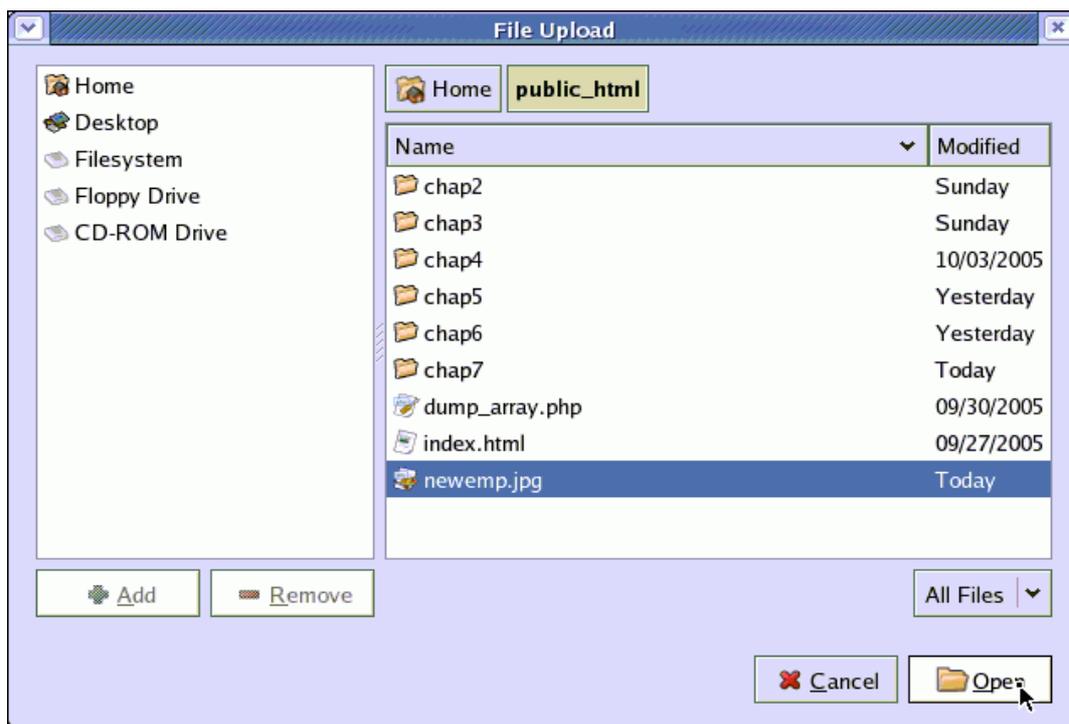
Insert New Employee

Department ID	<input type="text" value="10"/>
First Name	<input type="text" value="Chris"/>
Last Name	<input type="text" value="Jones"/>
Hiredate	<input type="text" value="11-OCT-05"/>
Job	<input type="text" value="Marketing Manager"/>
Salary	<input type="text" value="9000"/>
Commission (%)	<input type="text" value="0"/>
Photo	<input type="text"/> <input type="button" value="Browse..."/>

2005-10-11 12:32:04 Any Co.

[chap7_loading_020.gifの説明](#)

7. サイズが100ピクセルより大きいJPEGイメージを検索して選択し、「Open」をクリックします。



[chap7_loading_021.gifの説明](#)

8. 「Insert New Employee」 ページで、「Save」をクリックします。

Insert New Employee

Department ID	<input type="text" value="10"/>
First Name	<input type="text" value="Chris"/>
Last Name	<input type="text" value="Jones"/>
Hiredate	<input type="text" value="11-OCT-05"/>
Job	<input type="text" value="Marketing Manager"/>
Salary	<input type="text" value="9000"/>
Commission (%)	<input type="text" value="0"/>
Photo	<input type="text" value="/home/gstokol/public_html"/> <input type="button" value="Browse..."/>
<input type="button" value="Save"/> <input type="button" value="Cancel"/>	

2005-10-11 12:32:04 Any Co.

[chap7_loading_022.gifの説明](#)

「Employees」 ページに、新しくアップロードしたJPEGイメージが表示されます。そのイメージ・サイズは、イメージ・サイズ変更コードを含める前にロードしたイメージと比較すると小さくなります。

Employees: Administration

	Employee ID	Employee Name	Hiredate	Salary	Commission (%)	Remuneration	Photograph
	200	J. Whalen	17-SEP-87	4,400.00	0	52,800.00	Employee photo
	209	G. Stokol	11-OCT-05	8,000.00	0	96,000.00	
	210	C. Jones	11-OCT-05	9,000.00	0	108,000.00	

[chap7_loading_023.gifの説明](#)

8 グローバル・アプリケーションの構築

この章では、PHPおよびOracle Database環境でのグローバル・アプリケーションの開発について説明します。グローバルなインターネット・アプリケーションの開発およびデプロイに関連付けられている基本的なタスク（ロケール認識の開発、ユーザー選択言語でのHTMLコンテンツの構築、ユーザーのロケールの表記規則に従ったデータの表示など）について説明します。

様々なロケールがサポートされているグローバルなインターネット・アプリケーションを構築するには、優れた開発プラクティスが必要です。ロケールとは、各国語およびその言語が話されている地域のことです。アプリケーション自体がユーザーのロケール・プリファレンスを認識し、ユーザーが考えているとおりの表記規則に従ってコンテンツを表示する必要があります。適切なロケール特性（正しい日付書式や数値書式など）でデータを表示することが重要です。Oracle Databaseは、完全な国際化を実現しており、グローバル・アプリケーションを開発およびデプロイするためのグローバルなプラットフォームを提供します。

この章の内容は次のとおりです。

- [OracleとPHP間の環境の確立](#)
- [文字列の操作](#)
- [ユーザーのロケールの決定](#)
- [ロケール認識の開発](#)
- [HTMLページのエンコーディング](#)
- [翻訳のためのHTMLページのコンテンツの編成](#)
- [ユーザーが考えているとおりの表記規則を使用したデータの表示](#)

OracleとPHP間の環境の確立

グローバル・アプリケーションを構築するための最初の手順は、PHPエンジンとOracle Databaseの間に接続を正しく設定する手順です。これで、すべての層にわたってデータ整合性が保証されます。インターネット・ベースのほとんどの標準規格で、文字エンコーディングとしてUnicodeがサポートされているため、この章ではデータ交換用のキャラクタ・セットとしてUnicodeを使用します。

PHPはOCI8拡張モジュールを使用するため、OCIに適用されるルールはPHPにも適用されます。Oracleロケールの動作（OCIアプリケーションで 사용되는クライアント・キャラクタ・セットを含む）は、NLS_LANG環境変数で定義します。この環境変数の形式は次のとおりです。

```
<language>_<territory>.<character set>
```

たとえば、ドイツのドイツ語ユーザーがUnicodeでアプリケーションを実行している場合、NLS_LANGは次のように設定されています。

```
GERMAN_GERMANY.AL32UTF8
```

言語および地域の設定によって、Oracle日付書式、エラー・メッセージ言語、ソート順序に使用されるルールなどのOracleの動作が制御されます。キャラクタ・セットAL32UTF8は、UTF-8のOracle名です。

NLS_LANG環境変数の詳細は、Oracle Databaseのインストレーション・ガイドを参照してください。

PHPがApacheにインストールされている場合は、`/etc/profile`にNLS_LANGを設定できます。

```
export NLS_LANG GERMAN_GERMANY.AL32UTF8
```

PHPがOracle HTTP Serverにインストールされている場合は、\$ORACLE_HOME/opmn/conf/opmn.xmlに環境変数としてNLS_LANGを設定する必要があります。

```
<ias-component id="HTTP_Server">
<process-type id="HTTP_Server" module-id="OHS">
  <environment>
    <variable id="PERL5LIB"
      value="D:\oracle\1012J2EE\Apache\Apache\mod_perl\site\5.6.1\lib"/>
    <variable id="PHPRC" value="D:\oracle\1012J2EE\Apache\Apache\conf"/>
    <variable id="NLS_LANG" value="german_germany.al32utf8"/>
  </environment>
</module-data>
<category id="start-parameters">
  <data id="start-mode" value="ssl-disabled"/>
</category>
</module-data>
<process-set id="HTTP_Server" numprocs="1"/>
</process-type>
</ias-component>
```

この変更を実装するには、Webリスナーを再起動する必要があります。

文字列の操作

PHPは、ISO-8859-1キャラクタ・セットで動作するように設計されています。それ以外のキャラクタ・セット（特にマルチバイト・キャラクタ・セット）を処理する場合は、一連のマルチバイト文字列関数を使用できます。これらの機能を有効にするには、mbstring拡張モジュールを有効にする必要があります。

アプリケーション・コードでmb_strlen()などの関数を使用して、文字列内の文字数を計算する必要があります。これによって、文字列内のバイト数を戻すstrlen()とは異なる値が戻される場合があります。

mbstring拡張モジュールを有効にし、Webサーバーを再起動すると、いくつかの構成オプションを使用できるようになります。mbstring.func_overloadをOverload設定のいずれかに設定して、標準のPHP文字列関数の動作を変更できます。

詳細は、PHPのmbstringに関する次のリファレンス・マニュアルを参照してください。

<http://www.php.net/mbstring>

ユーザーのロケールの決定

グローバル環境のアプリケーションでは、ロケール・プリファレンスが異なるユーザーに対応する必要があります。ユーザーの優先ロケールを決定した後は、そのロケールの言語でHTMLコンテンツを構築し、そのロケールの表記規則に従います。

ユーザーのロケールを決定する一般的な方法として、ブラウザのデフォルトのISOロケール設定を使用する方法があります。通常、ブラウザは、Accept Language HTTPヘッダーでHTTPサーバーにそのロケール・プリファレンス設定を送信します。Accept Language HTTPヘッダーがNULLの場合は、使用可能なロケール・プリファレンス情報がなく、事前定義のデフォルトのロケールにフォールバックされます。

次のPHPコードは、\$_SERVERサーバー変数を使用して、Accept Language HTTPヘッダーからISOロケールを取り出します。

```
$s = $_SERVER["HTTP_ACCEPT_LANGUAGE"]
```

ロケール認識の開発

ユーザーのロケール・プリファレンスを決定すると、日付、時間、通貨の書式などのロケール依存の関数をコールして、ロケールの表記規則に従ってHTMLページを書式設定できます。

様々なプログラミング環境で実装されるグローバル・アプリケーションを構築する場合は、異なる環境間でユーザー・ロケール設定を同期できるようにする必要があります。たとえば、PL/SQLプロシージャをコールするPHPアプリケーションでは、PL/SQLプロシージャをコールする前に、ISOロケールに対応するNLS_LANGUAGE値およびNLS_TERRITORY値にマップし、ユーザーのロケールに合わせてパラメータ値を変更する必要があります。

ります。PL/SQL UTL_I18Nパッケージには、ISOロケールとOracleロケール間でマップできるマッピング関数が含まれています。

表8-1に、よく使用されるロケールをISO環境およびOracle環境に定義する方法を示します。

表 8-1: ISO、SQL、PL/SQLの各プログラミング環境でのロケールの表現

ロケール	ロケールID	NLS_LANGUAGE	NLS_TERRITORY
中国語（中華人民共和国）	zh-CN	SIMPLIFIED CHINESE	CHINA
中国語（台湾）	zh-TW	TRADITIONAL CHINESE	TAIWAN
英語（アメリカ合衆国）	en-US	AMERICAN	AMERICA
英語（イギリス）	en-GB	ENGLISH	UNITED KINGDOM
フランス語（カナダ）	fr-CA	CANADIAN FRENCH	CANADA
フランス語（フランス）	fr-FR	FRENCH	FRANCE
ドイツ語	de	GERMAN	GERMANY
イタリア語	it	ITALIAN	ITALY
日本語	ja	JAPANESE	JAPAN
韓国語	ko	KOREAN	KOREA
ポルトガル語（ブラジル）	pt-BR	BRAZILIAN PORTUGUESE	BRAZIL
ポルトガル語	pt	PORTUGUESE	PORTUGAL
スペイン語	es	SPANISH	SPAIN

HTMLページのエンコーディング

HTMLページのエンコーディングは、ブラウザおよびインターネット・アプリケーションに関する重要な情報です。ページ・エンコーディングは、インターネット・アプリケーションがサービスを提供しているロケールで使用されるキャラクタ・セットとみなすことができます。ブラウザは、正しいフォントおよびキャラクタ・セットのマッピング表を使用してHTMLページを表示できるように、ページ・エンコーディングを認識している必要があります。インターネット・アプリケーションは、HTMLフォームに入力されたデータを処理できるように、HTMLページ・エンコーディングを認識している必要があります。

ロケールごとに異なるネイティブ・エンコーディングを使用するのではなく、すべてのページ・エンコードでUTF-8 (Unicodeエンコーディング) を使用することをお勧めします。このエンコーディングを使用すると、グローバル・アプリケーションのコーディングを簡略化できるだけでなく、単一ページに多言語コンテンツを配置できます。

HTMLページ用のページ・エンコーディングの指定

HTMLページのエンコーディングをHTTPヘッダーまたはHTMLページ・ヘッダーに指定できます。

HTTPヘッダーへのエンコーディングの指定

HTMLページ・エンコーディングをHTTPヘッダーに指定するには、HTTP仕様にContent-Type HTTPヘッダーを含めます。このヘッダーで、コンテンツ・タイプおよびキャラクタ・セットを指定します。Content-Type HTTPヘッダーの形式は次のとおりです。

```
Content-Type: text/html; charset=utf-8
```

charsetパラメータで、HTMLページのエンコーディングを指定します。charsetパラメータに指定可能な値は、ブラウザでサポートされている文字エンコーディングのIANA名です。

HTMLページ・ヘッダーへのエンコーディングの指定

この方法は、主に静的なHTMLページに使用します。HTMLページ・エンコーディングをHTMLページ・ヘッダーに指定するには、文字エンコーディングを次のようにHTMLヘッダーに指定します。

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

charsetパラメータで、HTMLページのエンコーディングを指定します。Content-Type HTTPの場合と同様に、charsetパラメータに指定可能な値は、ブラウザでサポートされている文字エンコーディングのIANA名です。

PHPへのページ・エンコーディングの指定

PHP構成変数を次のように設定することで、Content-Type HTTPヘッダーでHTMLページのエンコーディングを指定できます。

```
default_charset = UTF-8
```

この設定によって送信ページは変換されません。アプリケーションでは、サーバー生成のページをUTF-8でエンコードする必要があります。

翻訳のためのHTMLページのコンテンツの編成

ユーザー・インタフェースをユーザーのローカル言語で使用できるようにすることは、アプリケーションのグローバル化の基本的なタスクです。HTMLページのコンテンツの翻訳可能なソースは、次のように分類されます。

- アプリケーション・コードに含まれているテキスト文字列
- 静的なHTMLファイル、イメージ・ファイル、およびCSSなどのテンプレート・ファイル
- データベースに格納されている動的なデータ

PHPの文字列

PHPアプリケーション・ロジック内の翻訳可能な文字列を外部化して、そのテキストを簡単に翻訳できるようにする必要があります。これらのテキスト・メッセージは、翻訳対象データのタイプおよびボリュームに応じて、フラット・ファイルまたはデータベース表に格納できます。

静的ファイル

HTMLファイルやGIFファイルなどの静的ファイルは簡単に翻訳できます。これらのファイルを翻訳する場合は、UTF-8をファイル・エンコーディングとして指定して、対応する言語に翻訳する必要があります。翻訳したファイルの言語を区別するには、静的ファイルを言語ごとに異なるディレクトリにステージングするか、または異なるファイル名でステージングします。

データベースのデータ

通常、製品名や製品説明などの動的な情報がデータベースに格納されています。様々な翻訳を区別するには、この情報を保持しているデータベース・スキーマに言語を示すための列を含める必要があります。目的の言語を選択するには、問合せにWHERE句を含める必要があります。

ユーザーが考えているとおりの表記規則を使用したデータの表示

アプリケーションのデータは、ユーザーが考えているとおりに表示される必要があります。そうでない場合、データの意味が誤って解釈される可能性があります。たとえば、12/11/05という日付は、アメリカ合衆国では2005年12月11日を意味するのに対して、イギリスでは2005年11月12日を意味します。ユーザーの数値書式および通貨書式にも同様の違いが存在します。たとえば、.という記号はアメリカ合衆国では小数点セパレータですが、ドイツでは千単位セパレータです。

各言語に独自のソート・ルールがあります。アルファベットの文字の順序に従う言語、文字の画数に従う言語、単語の発音に従う言語などがあります。ユーザーが慣れている言語順序でソートされていないデータを表示すると、情報の検索が難しくなり、時間がかかることがあります。

アプリケーション・ロジックおよびデータベースから取り出すデータの量によっては、アプリケーション・レベルではなくデータベース・レベルでデータを書式設定した方がより適切な場合があります。Oracle Databaseには、ユーザーのロケール・プリファレンスがわかっている場合にデータの表示を調整するのに役立つ機能が多く用意されています。以降の項では、SQLでのロケール依存の操作の例を示します。

Oracleの日付書式

Oracle Databaseの日付表示書式には、標準の日付、短い日付、長い日付の3種類があります。次の例では、アメリカ合衆国およびドイツでの短い日付書式と長い日付書式の違いを示します。

```
SQL> alter session set nls_territory=america nls_language=american;
```

Session altered.

```
SQL> select employee_id EmpID,
2 substr(first_name,1,1)||'.'||last_name "EmpName",
3 to_char(hire_date,'DS') "Hiredate",
4 to_char(hire_date,'DL') "Long HireDate"
5 from employees
6* where employee_id <105;
```

EMPID	EmpName	Hiredate	Long HireDate
100	S.King	06/17/1987	Wednesday, June 17, 1987
101	N.Kochhar	09/21/1989	Thursday, September 21, 1989
102	L.De Haan	01/13/1993	Wednesday, January 13, 1993
103	A.Hunold	01/03/1990	Wednesday, January 3, 1990
104	B.Ernst	05/21/1991	Tuesday, May 21, 1991

```
SQL> alter session set nls_territory=germany nls_language=german;
```

Session altered.

```
SQL> select employee_id EmpID,
2 substr(first_name,1,1)||'.'||last_name "EmpName",
3 to_char(hire_date,'DS') "Hiredate",
4 to_char(hire_date,'DL') "Long HireDate"
5 from employees
6* where employee_id <105;
```

EMPID	EmpName	Hiredate	Long HireDate
100	S.King	17.06.87	Mittwoch, 17. Juni 1987
101	N.Kochhar	21.09.89	Donnerstag, 21. September 1989
102	L.De Haan	13.01.93	Mittwoch, 13. Januar 1993
103	A.Hunold	03.01.90	Mittwoch, 3. Januar 1990
104	B.Ernst	21.05.91	Dienstag, 21. Mai 1991

Oracleの数値書式

次の例では、アメリカ合衆国とドイツでの小数点文字およびグループ・セパレータの違いを示します。

```
SQL> alter session set nls_territory=america;
```

Session altered.

```
SQL> select employee_id EmpID,
2  substr(first_name,1,1)||'|'||last_name "EmpName",
3  to_char(salary, '99G999D99') "Salary"
4  from employees
5* where employee_id <105
```

EMPID EmpName	Salary
100 S.King	24,000.00
101 N.Kochhar	17,000.00
102 L.De Haan	17,000.00
103 A.Hunold	9,000.00
104 B.Ernst	6,000.00

```
SQL> alter session set nls_territory=germany;
```

Session altered.

```
SQL> select employee_id EmpID,
2  substr(first_name,1,1)||'|'||last_name "EmpName",
3  to_char(salary, '99G999D99') "Salary"
4  from employees
5* where employee_id <105
```

EMPID EmpName	Salary
100 S.King	24.000,00
101 N.Kochhar	17.000,00
102 L.De Haan	17.000,00
103 A.Hunold	9.000,00
104 B.Ernst	6.000,00

Oracleの言語ソート

スペイン語では、伝統的に *ch*、*ll* および *ñ* が一意の文字として扱われ、それぞれ *c*、*l* および *n* の後に順序付けられます。次の例では、Chen と Chung という従業員名にスペイン語のソートを使用した場合の結果を示します。

```
SQL> alter session set nls_sort=binary;
```

Session altered.

```
SQL> select employee_id EmpID,
2  last_name "Last Name"
3  from employees
4  where last_name like 'C%'
5* order by last_name
```

EMPID Last Name
187 Cabrio
148 Cambrault
154 Cambrault
110 Chen
188 Chung
119 Colmenares

6 rows selected.

```
SQL> alter session set nls_sort=spanish_m;
```

Session altered.

```
SQL> select employee_id EmpID,
2  last_name "Last Name"
3  from employees
4  where last_name like 'C%'
5* order by last_name
```

EMPID Last Name

```
-----  
187 Cabrio  
148 Cambrault  
154 Cambrault  
119 Colmenares  
110 Chen  
188 Chung
```

6 rows selected.

Oracleのエラー・メッセージ

NLS_LANGUAGE パラメータは、データベースから戻されるデータベース・エラー・メッセージの言語も制御します。SQL文を発行する前にこのパラメータを設定すると、言語固有のデータベース・エラー・メッセージがアプリケーションに戻されるようになります。

次のサーバー・メッセージについて考えてみます。

ORA-00942: table or view does not exist

NLS_LANGUAGEパラメータをフランス語に設定すると、このサーバー・メッセージは次のように表示されます。

ORA-00942: table ou vue inexistante

Oracle Databaseのグローバリゼーション・サポート機能の詳細は、『Oracle Database 2日で開発者ガイド』のグローバル環境での作業に関する項を参照してください。

索引

記号 [A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [J](#) [L](#) [M](#) [N](#) [O](#) [P](#) [S](#) [T](#) [U](#) [V](#) [W](#) [あ](#) [か](#) [さ](#) [た](#) [は](#) [や](#) [ら](#)

記号

`$bindargs`配列, [\[1\]](#)
`$bindargs`パラメータ, [\[1\]](#)
`$bindvars`パラメータ, [\[1\]](#), [\[2\]](#), [\[3\]](#)
`$conn`パラメータ, [\[1\]](#)
`$current`変数, [\[1\]](#)
`$date`パラメータ, [\[1\]](#)
`$DID`変数, [\[1\]](#)
`$emp`変数, [\[1\]](#)
`$e`パラメータ, [\[1\]](#), [\[2\]](#), [\[3\]](#)
`$file`パラメータ, [\[1\]](#)
`$line`パラメータ, [\[1\]](#)
`$posturl`パラメータ, [\[1\]](#), [\[2\]](#)
`$q1`パラメータ [\[1\]](#)
`$query`パラメータ, [\[1\]](#)
`$refcur`変数, [\[1\]](#)
`$results`パラメータ, [\[1\]](#)
`$resulttype`パラメータ, [\[1\]](#)
`$rowsperpage`パラメータ, [\[1\]](#)
`$stid`パラメータ, [\[1\]](#), [\[2\]](#)
`$title`パラメータ, [\[1\]](#)
`$`パラメータ, [\[1\]](#)
@ (エラー表示を抑止する), [\[1\]](#)
__FILE__変数, [\[1\]](#), [\[2\]](#)
__LINE__変数, [\[1\]](#), [\[2\]](#)

A

AL32UTF8キャラクタ・セット, [\[1\]](#)
anyco.php
 DB接続の追加, [\[1\]](#)
 「Employees」ページ, [\[1\]](#)
 「Employees」ページおよび「Departments」ページ, [\[1\]](#)
 PL/SQL関クションのコール, [\[1\]](#)
 イメージの挿入, [\[1\]](#)
 インクルード・ファイル, [\[1\]](#)
 エラー処理, [\[1\]](#), [\[2\]](#)
 作成, [\[1\]](#)
 実行する問合せ, [\[1\]](#)
 従業員ではなく部門, [\[1\]](#)
 従業員レコードの更新, [\[1\]](#)
 従業員レコードの構築, [\[1\]](#)
 従業員レコードの削除, [\[1\]](#)
 従業員レコードの挿入, [\[1\]](#)
 使用可能なデータベース接続, [\[1\]](#)
 すべての行のフェッチ, [\[1\]](#)
 説明, [\[1\]](#)
 テスト, [\[1\]](#), [\[2\]](#), [\[3\]](#), [\[4\]](#), [\[5\]](#), [\[6\]](#), [\[7\]](#), [\[2\]](#), [\[3\]](#), [\[4\]](#), [\[5\]](#), [\[6\]](#), [\[7\]](#), [\[8\]](#), [\[9\]](#)

データ操作ロジック, [\[1\]](#)
デフォルト部門の取得, [\[1\]](#)
問合せの置換え, [\[1\]](#)
問合せの実行, [\[1\]](#)
ナビゲーション, [\[1\]](#)
バインド変数, [\[1\]](#)
バインド変数の使用, [\[1\]](#)
ページ・タイトルの出力, [\[1\]](#)

AnyCo Corp

チュートリアル・アプリケーション, [\[1\]](#)

AnyCo Corpのチュートリアル, [\[1\]](#)
anyco_cn. inc

作成, [\[1\]](#)
説明, [\[1\]](#)

anyco_db. inc

anyco.phpへのインクルード, [\[1\]](#)
PL/SQLパッケージ・プロシージャのコール, [\[1\]](#)
エラーの戻し, [\[1\]](#)
エラー・パラメータの受渡し, [\[1\]](#)
エラー表示の抑止, [\[1\]](#)
エラー変数の戻し, [\[1\]](#)
作成, [\[1\]](#)
サブセットの問合せ, [\[1\]](#)
サムネイル・イメージの作成, [\[1\]](#)
サムネイル・イメージの挿入, [\[1\]](#)
出力タイプの選択, [\[1\]](#), [\[2\]](#)
すべての関数からのエラーの戻し, [\[1\]](#)
説明, [\[1\]](#)
テスト, [\[1\]](#)

データ操作文の実行, [\[1\]](#)
バインド変数のコール, [\[1\]](#)
バインド変数の変更, [\[1\]](#)
anyco_im.php

作成, [\[1\]](#)
従業員イメージの表示, [\[1\]](#)

説明, [\[1\]](#)
anyco_ui. inc

anyco.phpへのインクルード, [\[1\]](#)
HTML表の従業員データ, [\[1\]](#)
インクルード, [\[1\]](#)
エラーの出力, [\[1\]](#)
拡張, [\[1\]](#)
関数, [\[1\]](#)
結果の書式設定, [\[1\]](#)
作成, [\[1\]](#), [\[2\]](#)
従業員imgタグの生成, [\[1\]](#)
従業員イメージのアップロード, [\[1\]](#)

従業員イメージの列の追加, [\[1\]](#)
従業員データ用HTMLフォームの生成, [\[1\]](#)
従業員データ用のHTMLフォーム, [\[1\]](#)
従業員の報酬の列の追加, [\[1\]](#)
従業員レコードの更新, [\[1\]](#)
説明, [\[1\]](#)
ナビゲーションの追加, [\[1\]](#)
変更のテスト, [\[1\]](#)
Apache

httpd.conf構成ファイル, [\[1\]](#)
Linuxでのインストールのテスト, [\[1\]](#)
public_html, [\[1\]](#)
public_htmlの作成, [\[1\]](#)
Windowsでのインストールのテスト [\[1\]](#)

再起動, [\[1\]](#)
Apacheの再起動, [\[1\]](#)
Apacheのパブリック仮想ディレクトリ, [\[1\]](#)
array_push()関数, [\[1\]](#)

B

BLOB

従業員イメージの格納, [\[1\]](#)

C

calc_remuneration()関数, [\[1\]](#)
charsetパラメータ, [\[1\]](#)
construct_departments()関数, [\[1\]](#), [\[2\]](#), [\[3\]](#), [\[4\]](#), [\[5\]](#)
construct_employees()関数, [\[1\]](#), [\[2\]](#), [\[3\]](#), [\[4\]](#)
construct_image()関数, [\[1\]](#)
construct_insert_emp()関数, [\[1\]](#), [\[2\]](#)
construct_modify_emp()関数, [\[1\]](#)
Content-type, [\[1\]](#)
COUNTRIES表, [\[1\]](#)
cv_types.et_employees()プロシージャ, [\[1\]](#)

D

date()関数, [\[1\]](#)
db_do_query()関数, [\[1\]](#), [\[2\]](#), [\[3\]](#), [\[4\]](#), [\[5\]](#), [\[6\]](#), [\[7\]](#), [\[8\]](#), [\[9\]](#)
db_error()関数, [\[1\]](#), [\[2\]](#), [\[3\]](#), [\[4\]](#)
db_execute_statement()関数, [\[1\]](#), [\[2\]](#), [\[3\]](#)
db_get_employees_rc()関数, [\[1\]](#)
db_get_page_data()関数, [\[1\]](#), [\[2\]](#), [\[3\]](#), [\[4\]](#), [\[5\]](#)
db_insert_thumbnail()関数, [\[1\]](#), [\[2\]](#), [\[3\]](#)
delete_emp()関数, [\[1\]](#)
DEPARTMENTS表, [\[1\]](#), [\[2\]](#), [\[3\]](#), [\[4\]](#)

「Departments」 ページ
「Employees」 との結合, [\[1\]](#)
拡張, [\[1\]](#)

department_id変数, [\[1\]](#)
deptidパラメータ, [\[1\]](#)
DIDバインド変数, [\[1\]](#), [\[2\]](#), [\[3\]](#)
display_errorsディレクティブ, [\[1\]](#)
do_query()関数, [\[1\]](#)

E

EMPLOYEES表, [\[1\]](#), [\[2\]](#), [\[3\]](#), [\[4\]](#), [\[5\]](#), [\[6\]](#), [\[7\]](#)
「Employees」 ページ, [\[1\]](#)
「Departments」 との結合, [\[1\]](#)
拡張, [\[1\]](#)
作成, [\[1\]](#)

EMPLOYEE_PHOTOS表, [\[1\]](#), [\[2\]](#)
enctype属性, [\[1\]](#)
error_reportingディレクティブ, [\[1\]](#)
false文の戻し, [\[1\]](#)

F

FIRSTバインド変数, [\[1\]](#)

G

GDグラフィックス拡張モジュール, [\[1\]](#)
get_dept_name()関数, [\[1\]](#)

H

header()関数, [\[1\]](#)
hello.php
PHPインストールのテスト, [\[1\]](#)

HRアカウントのロック解除, [\[1\]](#), [\[2\]](#)
HTML

カスケード・スタイルシート, [\[1\]](#)
従業員データの出力, [\[1\]](#)
従業員データ用フォームの生成, [\[1\]](#)
従業員データを含むフォーム, [\[1\]](#)
ページ・エンコーディング, [\[1\]](#), [\[2\]](#), [\[3\]](#), [\[4\]](#)
ページ・ヘッダー, [\[1\]](#)

htmlentities()関数, [\[1\]](#)
httpd.conf Apache構成ファイル, [\[1\]](#)
HTTPヘッダー

ページ・エンコーディング, [\[1\]](#)

I

imagecopyresampled()関数, [\[1\]](#)
imagecreatefromjpeg()関数, [\[1\]](#)
imagecreatetruecolor()関数, [\[1\]](#)
insert_new_emp()関数, [\[1\]](#), [\[2\]](#), [\[3\]](#)
isset()関数, [\[1\]](#)

J

JOBS表, [\[1\]](#)
JPEGファイル, [\[1\]](#)

L

LASTバインド変数, [\[1\]](#)
LOCATIONS表, [\[1\]](#)

M

modify_emp()関数, [\[1\]](#)

N

NEWIDバインド変数, [\[1\]](#)
NLS_LANGUAGE環境変数, [\[1\]](#), [\[2\]](#)
NLS_LANG環境変数, [\[1\]](#)
NLS_TERRITORY環境変数, [\[1\]](#)
NULL値, [\[1\]](#)

[OCI-Lob->load\(\)関数](#), [\[1\]](#)
[oci8.max_persistent設定](#), [\[1\]](#)
[oci8.persistent_timeout設定](#), [\[1\]](#)
[oci8.ping_interval設定](#), [\[1\]](#)
[OCI8 oci_pconnect\(\)](#), [\[1\]](#)
[OCI8 oci_pconnect\(\)関数](#), [\[1\]](#)
[oci_bind_by_name\(\)関数](#), [\[1\]](#), [\[2\]](#)
[OCI_B_CURSOR参照カーソル](#), [\[1\]](#)
[oci_close\(\)関数](#), [\[1\]](#)
[OCI_COMMIT_ON_SUCCESSパラメータ](#), [\[1\]](#)
[oci_connect\(\)関数](#), [\[1\]](#), [\[2\]](#), [\[3\]](#)
[OCI_DEFAULTパラメータ](#), [\[1\]](#)
[oci_error\(\)関数](#), [\[1\]](#)
[oci_execute\(\)関数](#), [\[1\]](#)
[OCI_FETCHSTATEMENT_BY_COLUMNパラメータ](#), [\[1\]](#)
[OCI_FETCHSTATEMENT_BY_ROWパラメータ](#), [\[1\]](#), [\[2\]](#), [\[3\]](#)
[oci_fetch_all\(\)関数](#), [\[1\]](#), [\[2\]](#)
[oci_fetch_array\(\)関数](#), [\[1\]](#)
[oci_parse\(\)関数](#), [\[1\]](#)
[OCI_RETURN_NULLSパラメータ](#), [\[1\]](#)
 Oracle
 [tnsnames.ora](#), [\[1\]](#)
 環境の確立, [\[1\]](#)
 数値書式, [\[1\]](#)
 日付書式, [\[1\]](#)

Oracle Database

[インストール](#), [\[1\]](#)
 取得およびインストール, [\[1\]](#)
 接続, [\[1\]](#)
 前提条件, [\[1\]](#)

Oracle Databaseの前提条件, [\[1\]](#)
 Oracleの数値書式, [\[1\]](#)
 Oracleの日付書式, [\[1\]](#)
 OUTバインド変数, [\[1\]](#)

PHP
 [display_errorsディレクティブ](#), [\[1\]](#)
 [error_reportingディレクティブ](#), [\[1\]](#)
 [GDグラフィックス拡張モジュール](#), [\[1\]](#), [\[2\]](#)
 [hello.php](#), [\[1\]](#)
 [HTMLおよびGIFファイルの翻訳](#), [\[1\]](#)
 HTMLページ・エンコーディング, [\[1\]](#)
 Linuxでのインストール, [\[1\]](#)
 NULL値, [\[1\]](#)
[oci8.max_persistent](#), [\[1\]](#)
[oci8.persistent_timeout](#), [\[1\]](#)
[oci8.ping_interval](#), [\[1\]](#)
 Windowsでのインストール, [\[1\]](#), [\[2\]](#), [\[3\]](#)
 アプリケーションのグローバル化, [\[1\]](#)
 アプリケーション・ロジック, [\[1\]](#)
 カスケード・スタイルシート, [\[1\]](#)

環境の確立, [\[1\]](#)
キャラクタ・セット, [\[1\]](#)
ヒア・ドキュメント, [\[1\]](#)
ファイルの作成, [\[1\]](#)
翻訳可能な文字列の外部化, [\[1\]](#)
ユーザー・ロケールの決定, [\[1\]](#)

PHP (PHP Hypertext Preprocessor) , [\[1\]](#)
PHP関数

 ui_print_footer(), [\[1\]](#)
ui_print_header(), [\[1\]](#)

PL/SQL

 anyco.phpのファンクションのコール, [\[1\]](#)
cv_types.et_employees()パッケージ・プロシージャ, [\[1\]](#)
UTL_I18Nパッケージ, [\[1\]](#)

アプリケーションのプロシージャおよびファンクション, [\[1\]](#)
パッケージ・プロシージャのコール, [\[1\]](#)
public_html

 Apache, [\[1\]](#)
作成, [\[1\]](#)

S

session_start()関数, [\[1\]](#)
style.css
 HTMLの表示, [\[1\]](#)

style.cssの説明, [\[1\]](#)
SYSDATE関数, [\[1\]](#)

T

tnsnames.ora, [\[1\]](#)

U

ui_print_department()関数, [\[1\]](#), [\[2\]](#), [\[3\]](#)
ui_print_employees()関数, [\[1\]](#), [\[2\]](#), [\[3\]](#), [\[4\]](#), [\[5\]](#), [\[6\]](#)
ui_print_error()関数, [\[1\]](#)
ui_print_footer()関数, [\[1\]](#)
ui_print_header()関数, [\[1\]](#)
ui_print_insert_employee()関数, [\[1\]](#), [\[2\]](#), [\[3\]](#)
ui_print_modify_employee()関数, [\[1\]](#)

UTF-8
HTMLページ・エンコーディング, [\[1\]](#)
キャラクタ・セット, [\[1\]](#)

UTL_I18Nパッケージ, [\[1\]](#)

V

var_dump()関数, [\[1\]](#), [\[2\]](#)

W

Webブラウザ
LinuxでのApacheインストールのテスト, [\[1\]](#)
WindowsでのApacheインストールのテスト, [\[1\]](#)

あ

アプリケーション, [\[1\]](#)
BLOBの従業員イメージ, [\[1\]](#)
「Departments」と「Employees」の結合, [\[1\]](#)
「Departments」ページの拡張, [\[1\]](#)
「Employees」ページの拡張, [\[1\]](#)
「Employees」ページの作成, [\[1\]](#)
HTMLおよびGIFの翻訳, [\[1\]](#)
「Next」および「Previous」ボタン, [\[1\]](#)
Oracleエラー, [\[1\]](#)
PL/SQLプロシージャおよびファンクション, [\[1\]](#)
UTF-8ページ・エンコーディング, [\[1\]](#)
エラーの出力, [\[1\]](#)
エラー変数の戻し, [\[1\]](#)
エラー・リカバリ, [\[1\]](#)
グローバル化, [\[1\]](#)
サブセットの問合せの実装, [\[1\]](#)
サムネイル・イメージ, [\[1\]](#)
実行する問合せ, [\[1\]](#)
従業員imgタグの生成, [\[1\]](#)
従業員イメージのアップロード, [\[1\]](#)
従業員イメージの追加, [\[1\]](#)
従業員イメージの表示, [\[1\]](#)
従業員サムネイルの挿入, [\[1\]](#)
従業員ではなく部門のコール, [\[1\]](#)
従業員の報酬の列の追加, [\[1\]](#)
従業員レコードの更新, [5.2](#)
従業員レコードの構築, [\[1\]](#)
従業員レコードの削除, [\[1\]](#)
従業員レコードの挿入, [\[1\]](#)
出力タイプの選択, [\[1\]](#), [\[2\]](#)
すべての行のフェッチ, [\[1\]](#)
接続関数, [\[1\]](#)
データ操作文の実行, [\[1\]](#)

データベース接続の定数, [\[1\]](#)
データベース・ロジックの集中化, [\[1\]](#)
デフォルト部門の取得, [\[1\]](#)
問合せの実行, [\[1\]](#)
ファイル・ディレクトリ, [\[1\]](#), [\[2\]](#)
ファイルのネーミング規則, [\[1\]](#)
ページ・タイトルの出力, [\[1\]](#)
翻訳可能な文字列の外部化, [\[1\]](#)
ユーザー・インタフェース, [\[1\]](#)
ユーザー・インタフェースの翻訳, [\[1\]](#)
レポート・ページ, [\[1\]](#)
ロケール固有の関数のコール, [\[1\]](#)

アプリケーションでのレポート, [\[1\]](#)
イメージ

anyco.phpへの挿入, [\[1\]](#)
BLOBでの格納, [\[1\]](#)

サムネイルの作成, [\[1\]](#)
サムネイルの作成, [7.2](#)
従業員サムネイルの挿入, [\[1\]](#)
インクルード・ファイル

anyco.php, [\[1\]](#)
anyco.php, [\[1\]](#), [\[2\]](#)

anyco_ui.inc, [\[1\]](#)
インストール

LinuxでのPHP, [\[1\]](#)
Oracle Database, [\[1\]](#), [\[2\]](#)
WindowsでのPHP, [\[1\]](#), [\[2\]](#), [\[3\]](#)

エラー

NLS_LANGUAGE, [\[1\]](#)
Oracle, [\[1\]](#)
処理, [\[1\]](#)
すべての関数からの戻し, [\[1\]](#)

パラメータの受渡し, [\[1\]](#)
表示の抑止, [\[1\]](#)
変数への割当て, [\[1\]](#)
戻し, [\[1\]](#)
リカバリ, [\[1\]](#)
エラーの戻し, [\[1\]](#)

か

簡易接続構文, [\[1\]](#)
環境変数
NLS_LANG, [\[1\]](#)
NLS_LANGUAGE, [\[1\]](#), [\[2\]](#)
NLS_TERRITORY, [\[1\]](#)

関数

[anycoui.inc](#), [\[1\]](#)
[array_push\(\)](#), [\[1\]](#)
[calc_remuneration\(\)](#), [\[1\]](#)
[construct_departments\(\)](#), [\[1\]](#), [\[2\]](#), [\[3\]](#), [\[4\]](#), [\[5\]](#)
[construct_employees\(\)](#), [\[1\]](#), [\[2\]](#), [\[3\]](#), [\[4\]](#)
[construct_image\(\)](#), [\[1\]](#)
[construct_insert_emp\(\)](#), [\[1\]](#), [\[2\]](#)
[construct_modify_emp\(\)](#), [\[1\]](#)
[date\(\)](#), [\[1\]](#)
[db_do_query\(\)](#), [\[1\]](#), [\[2\]](#), [\[3\]](#), [\[4\]](#), [\[5\]](#), [\[6\]](#), [\[7\]](#), [\[8\]](#), [\[9\]](#)
[db_error\(\)](#), [\[1\]](#), [\[2\]](#), [\[3\]](#), [\[4\]](#)
[db_execute_statement\(\)](#), [\[1\]](#), [\[2\]](#), [\[3\]](#)
[db_get_employees_rc\(\)](#), [\[1\]](#)
[db_get_page_data\(\)](#), [\[1\]](#), [\[2\]](#), [\[3\]](#), [\[4\]](#), [\[5\]](#)
[db_insert_thumbnail\(\)](#), [\[1\]](#), [\[2\]](#), [\[3\]](#)
[delete_emp\(\)](#), [\[1\]](#)
[do_query\(\)](#), [\[1\]](#)
[get_dept_name\(\)](#), [\[1\]](#)
[header\(\)](#), [\[1\]](#)
[htmlentities\(\)](#), [\[1\]](#)
[imagecopyresampled\(\)](#), [\[1\]](#)
[imagecreatefromjpeg\(\)](#), [\[1\]](#)
[imagecreatetruecolor\(\)](#), [\[1\]](#)
[insert_new_emp\(\)](#), [\[1\]](#), [\[2\]](#), [\[3\]](#)
[isset\(\)](#), [\[1\]](#)
[modify_emp\(\)](#), [\[1\]](#)
[OCI-Lob->load\(\)](#), [\[1\]](#)
[OCI8 oci_pconnect\(\)](#), [\[1\]](#)
[OCI8 oci_pconnect\(\)関数](#), [\[1\]](#)
[oci_bind_by_name\(\)](#), [\[1\]](#), [\[2\]](#)
[oci_close\(\)](#), [\[1\]](#)
[oci_connect\(\)](#), [\[1\]](#), [\[2\]](#), [\[3\]](#)
[oci_error\(\)](#), [\[1\]](#)
[oci_execute\(\)](#), [\[1\]](#)
[oci_fetch_all\(\)](#), [\[1\]](#), [\[2\]](#)
[oci_fetch_array\(\)](#), [\[1\]](#)
[oci_parse\(\)](#), [\[1\]](#)
[session_start\(\)](#), [\[1\]](#)
[SYSDATE](#), [\[1\]](#)
[ui_print_department\(\)](#), [\[1\]](#), [\[2\]](#), [\[3\]](#)
[ui_print_employees\(\)](#), [\[1\]](#), [\[2\]](#), [\[3\]](#), [\[4\]](#), [\[5\]](#), [\[6\]](#)
[ui_print_error\(\)](#), [\[1\]](#)
[ui_print_footer\(\)](#), [\[1\]](#)
[ui_print_header\(\)](#), [\[1\]](#)
[ui_print_insert_employee\(\)](#), [\[1\]](#), [\[2\]](#), [\[3\]](#)
[ui_print_modify_employee\(\)](#), [\[1\]](#)
[var_dump\(\)](#), [\[1\]](#), [\[2\]](#)

キャラクタ・セット

[AL32UTF8](#), [\[1\]](#)
[UTF-8](#), [\[1\]](#)

[グローバル化設定](#), [\[1\]](#)
グローバル化

[HTMLページ・エンコーディング](#), [\[1\]](#)

NLS_LANGUAGE, [\[1\]](#)
PHPおよびOracle環境, [\[1\]](#)
アプリケーション, [\[1\]](#)
キャラクタ・セット, [\[1\]](#)
言語ソート, [\[1\]](#)
数値書式, [\[1\]](#)
データのソート, [\[1\]](#)
データの表示, [\[1\]](#)
動的な情報, [\[1\]](#)
日付書式, [\[1\]](#)
ユーザー・インタフェースの翻訳, [\[1\]](#)

ユーザー・ロケールの決定, [\[1\]](#)
ロケール固有の関数のコール, [\[1\]](#)
言語ソート, [\[1\]](#)
構成

Apache httpd.conf, [\[1\]](#)

さ

作成

anyco_ui.incアプリケーション・ユーザー・インタフェース, [\[1\]](#)
PHPファイル, [\[1\]](#)
public_html, [\[1\]](#)
アプリケーション・ファイルのディレクトリ, [\[1\]](#), [\[2\]](#)

サムネイル・イメージ, [\[1\]](#)
参照カーソル

OCI_B_CURSOR, [\[1\]](#)

取得

Oracle Database, [\[1\]](#)

書式

anyco_ui.incの関数, [\[1\]](#)

人事管理 (HR) アプリケーション, [\[1\]](#)
接続

HRユーザー, [\[1\]](#), [\[2\]](#)
Oracle Database, [\[1\]](#)
永続, [\[1\]](#)
簡易接続構文, [\[1\]](#)
切断, [\[1\]](#)
設定, [\[1\]](#)

切断, [\[1\]](#)
ソート, [\[1\]](#), [\[2\]](#)

た

ディレクティブ

display_errors, [\[1\]](#)
error_reporting, [\[1\]](#)

テスト

anyco_db.inc, [\[1\]](#)
anyco_ui.inc, [\[1\]](#)
LinuxでのApacheインストール, [\[1\]](#)
WindowsでのApacheインストール, [\[1\]](#)

データベース

tnsnames.ora, [\[1\]](#)
簡易接続構文, [\[1\]](#)
従業員イメージの格納, [\[1\]](#)
すべての行のフェッチ, [\[1\]](#)
接続関数, [\[1\]](#)
接続の検証, [\[1\]](#)

接続の定数, [\[1\]](#)
切断, [\[1\]](#)
問合せの実行, [\[1\]](#)
動的な情報, [\[1\]](#)
レコードのナビゲート, [\[1\]](#)
ロジックの集中化, [\[1\]](#)
データベース・レコードのナビゲート, [\[1\]](#), [\[2\]](#)
デバッグ, [\[1\]](#)

は

バインド変数, [\[1\]](#), [\[2\]](#)

anyco.php, [\[1\]](#)
anyco_db.incのコール, [\[1\]](#)
DID, [\[1\]](#), [\[2\]](#), [\[3\]](#)
FIRST, [\[1\]](#)
LAST, [\[1\]](#)
NEWEID, [\[1\]](#)
OUT, [\[1\]](#)
データベースからの戻し, [\[1\]](#)

問合せの変更, [\[1\]](#)

バインド変数を使用した問合せ, [\[1\]](#)

\$bindargs, [\[1\]](#)
\$bindvars, [\[1\]](#), [\[2\]](#), [\[3\]](#)
\$conn, [\[1\]](#)
\$date, [\[1\]](#)
\$e, [\[1\]](#), [\[2\]](#), [\[3\]](#)
\$file, [\[1\]](#)
\$line, [\[1\]](#)
\$posturl, [\[1\]](#), [\[2\]](#)

[\\$q1, \[1\]](#)
[\\$query, \[1\]](#)
[\\$r, \[1\]](#)
[\\$results, \[1\]](#)
[\\$resulttype, \[1\]](#)
[\\$rowsperpage, \[1\]](#)
[\\$stid, \[1\], \[2\]](#)
[\\$title, \[1\]](#)

パラメータ

[charset, \[1\]](#)
[deptid, \[1\]](#)
[OCI_COMMIT_ON_SUCCESS, \[1\]](#)
[OCI_DEFAULT, \[1\]](#)
[OCI_FETCHSTATEMENT_BY_COLUMN, \[1\]](#)
[OCI_FETCHSTATEMENT_BY_ROW, \[1\], \[2\], \[3\]](#)
[OCI_RETURN_NULLS, \[1\]](#)

表

[COUNTRIES, \[1\]](#)
[DEPARTMENTS, \[1\], \[2\], \[3\], \[4\]](#)
[EMPLOYEES, \[1\], \[2\], \[3\], \[4\], \[5\], \[6\], \[7\]](#)
[EMPLOYEE_PHOTOS, \[1\], \[2\]](#)
[JOBS, \[1\]](#)
[LOCATIONS, \[1\]](#)

表記規則

[データの表示, \[1\]](#)

ファイル

[anyco.phpでのエラー処理, \[1\], \[2\]](#)
[anyco.phpの「Employees」ページ, \[1\]](#)
[anyco.phpのインクルード・ファイル, \[1\]](#)
[anyco.phpの作成, \[1\]](#)
[anyco.phpの説明, \[1\]](#)
[anyco.phpのテスト, \[1\], \[2\], \[3\], \[4\], \[5\], \[6\], \[7\], \[8\], \[9\], \[10\], \[11\], \[12\], \[13\], \[14\], \[15\]](#)
[anyco.phpへのDB接続の追加, \[1\]](#)
[anyco.phpへのデータ操作ロジックの追加, \[1\]](#)
[anyco.phpへのナビゲーションの追加, \[1\]](#)
[anyco_cn.incの作成\[1\]](#)
[anyco_cn.incの説明, \[1\]](#)
[anyco_db.incの作成\[1\]](#)
[anyco_db.incの説明, \[1\]](#)
[anyco_im.phpの作成, \[1\]](#)
[anyco_im.phpの説明, \[1\]](#)
[anyco_ui.incの拡張, \[1\]](#)
[anyco_ui.incの作成, \[1\]](#)
[anyco_ui.incの説明, \[1\]](#)
[anyco_ui.incのテスト, \[1\]](#)
[anyco_ui.incへのファイルのインクルード, \[1\]](#)
[「Employees」ページおよび「Departments」ページ, \[1\]](#)
[HTMLおよびGIFの翻訳, \[1\]](#)

[JPEG, \[1\]](#)

style.cssの説明, [\[1\]](#)
アプリケーション, [\[1\]](#), [\[2\]](#)
アプリケーションのネーミング規則, [\[1\]](#)
文, falseを戻す, [\[1\]](#)

\$current, [\[1\]](#)
\$DID, [\[1\]](#)
\$emp, [\[1\]](#)
\$refcur, [\[1\]](#)

変数

department_id, [\[1\]](#)

__FILE__, [\[1\]](#), [\[2\]](#)
__LINE__, [\[1\]](#), [\[2\]](#)

や

ユーザー・インタフェース
翻訳, [\[1\]](#)
翻訳可能な文字列の外部化, [\[1\]](#)

ら

ロケール, [\[1\]](#)

chap1_phpapp_001.gifは、6個のテキスト・ボックスを2列に3つずつ示しています。各ボックスは、チュートリアル・アプリケーションのファイルを表しています。いずれのボックスにも、各ファイルの説明が記載されています。各ボックスは矢印の付いた線で結ばれています。

左側の列には、上から順に次のボックスがあります。

anyco.php

ページ・リクエストを管理するためのフォーム・ハンドラ（コントローラ）が提供されている初期ページ。

anyco_ui.inc

ユーザー・インタフェース・ロジック：HTMLフォームおよびデータベース・データをHTML表に表示するPHPコードが記述されているインクルード・ファイル。anyco.phpを指している線があります。別の線がstyle.cssから指されています。3つ目の破線がanyco_im.phpを指しています。

style.css

HTML表示を管理するカスケード・スタイルシート（CSS）・ファイル。

右側の列には、上から順に次のボックスがあります。

anyco_cn.inc

デフォルトのデータベース接続情報を定義するインクルード・ファイル。ユーザー名、パスワードおよびデータベース接続識別子が記述されています。anyco.phpを指している線があります。

anyco_db.inc

モデル・ロジック：データベース接続を作成し、問合せおよびデータ操作文を実行するデータベース・ロジックが記述されているインクルード・ファイル。anyco.phpを指している線があります。

anyco_im.php

タグで参照されるPHPスクリプトで、指定した従業員のデータベース列からイメージを取得し、ブラウザに送信して表示します。

chap2_unlock.gifは、SQL DeveloperのSQLワークシートを示しています。

「Enter SQL Statement」 ペインには、次のテキストが含まれています。

```
alter user hr account unlock;
```

chap2_install_013.gifは、ブラウザに表示されたApacheテスト・ページを示しています。ページ最上部の段落には、次のテキストがあります。

If you can see this, it means that the installation of the Apache HTTP Server software on this system was successful. You may now add content to this directory and replace this page.

この段落のすぐ下に、水平線の区切りに続いて、「Seeing this instead of the website you expected?」というヘッダーがあります。その下に次の段落があります。

This page is here because the site administrator has changed the configuration of this web server. Please contact the person responsible for maintaining this server with questions. The Apache Software Foundation, which wrote the web server software this site administrator is using, has nothing to do with maintaining this site and cannot help resolve configuration issues.

この段落のすぐ下に、もう1つの水平線の区切りに続いて、次の段落があります。

The Apache documentation has been included with this distribution. (documentationがハイパーリンクされています)

You are free to use the image below on an Apache-powered web server. Thanks for using Apache!

この下に、次の語を含むロゴ・イメージがあります。

Powered by Apache

chap2_install_001.gifは、ブラウザに表示されたRed Hat Enterprise Linuxテスト・ページを示しています。ページのタイトルは「Red Hat Enterprise Linux Test Page」です。タイトルのすぐ下に次の段落があります。

This page is used to test the proper operation of the Apache HTTP server after it has been installed. If you can read this page, it means that the Apache HTTP server installed at this site is working properly.

次のヘッダーの下に、この後実行できる操作を示すコメントがあります。

- If you are a member of the general public
- If you are the website administrator

chap2_install_002.gifは、端末ウィンドウを示しています。手順3の説明に従って端末に入力したコマンドが表示されています。カーソルは新しい行上にあります。

chap2_install_003.gifは、端末ウィンドウを示しています。手順4の説明に従って端末に入力したコマンドが表示されています。

chap2_hello_001.gifは、「Hello, world!」というテキストが表示されているブラウザの画面を示しています。

chap3_test_install_005.gifは、「Departments」というタイトルのWebページを示しています。すぐ下に、ページの本体とヘッダーおよびフッターを区切る2本の水平線があります。本体には何も含まれていません。

ページのフッターの左側には日付2005-09-27および時間12:55:02、右側にはAny Co.が表示されています。

chap3_db_connect_001.gifは、「Departments」というタイトルのWebページを示しています。すぐ下には水平線があります。次に、8行4列の表があります。列にヘッダーはありません。各列に、次のデータが含まれています。

「10」、 「Administration」、 「200」、 「1700」

「20」、 「Marketing」、 「201」、 「1800」

「30」、 「Purchasing」、 「114」、 「1700」

「40」、 「Human Resources」、 「203」、 「2400」

「50」、 「Shipping」、 「121」、 「1500」

「60」、 「IT」、 「103」、 「1400」

「70」、 「Public Relations」、 「204」、 「2700」

「80」、 「Sales」、 「145」、 「2500」

chap4_db_connect_002.gifは、「Departments」というタイトルのWebページを示しています。すぐ下に、ヘッダーとページの本体を区切る水平線があります。次に、2行4列の表があります。最初の行には、次の列ヘッダーが含まれています。

「Department ID」、「Department Name」、「Manager ID」、「Location ID」

データ行には、次のデータが含まれています。

「80」、「Sales」、「145」、「2500」

すぐ下に、ページの本体とフッターを区切る水平線があります。フッターの左側には日付2005-09-30および時間11:50:00、右側にはAny Co.が表示されています。

chap4_db_connect_003.gifは、「Departments」というタイトルのWebページを示しています。すぐ下に、ヘッダーとページの本体を区切る水平線があります。次に、2行4列の表があります。最初の行には、次の列ヘッダーが含まれています。

「Department ID」、「Department Name」、「Manager ID」、「Location ID」

データ行には、次のデータが含まれています。

「80」、「Sales」、「145」、「2500」

すぐ下に、ページの本体とフッターを区切る水平線があります。フッターの左側には日付2005-09-30および時間14:42:50、右側にはAny Co.が表示されています。

chap4_db_nagivation_001.gifは、「Department」というタイトルのWebページを示しています。すぐ下に、ヘッダーとページの本体を区切る水平線があります。次に、2行4列の表があります。最初の行には、次の列ヘッダーが含まれています。

「Department ID」、「Department Name」、「Manager ID」、「Location ID」

データ行には、次のデータが含まれています。

「10」、「Administration」、「200」、「1700」

これらのデータの下には、「<Previous」および「Next>」というラベルの付いた2つのボタンがあります。マウス・カーソルは、「Next>」ボタンを指しています。

すぐ下に、ページの本体とフッターを区切る水平線があります。フッターの左側には日付2005-10-02および時間22:58:19、右側にはAny Co.が表示されています。

chap4_db_nagivation_002.gifは、「Department」というタイトルのWebページを示しています。すぐ下に、ヘッダーとページの本体を区切る水平線があります。次に、2行4列の表があります。最初の行には、次の列ヘッダーが含まれています。

「Department ID」、「Department Name」、「Manager ID」、「Location ID」

データ行には、次のデータが含まれています。

「20」、「Marketing」、「201」、「1800」

これらのデータの下には、「<Previous」および「Next>」というラベルの付いた2つのボタンがあります。マウス・カーソルは、「<Previous」ボタンを指しています。

すぐ下に、ページの本体とフッターを区切る水平線があります。フッターの左側には日付2005-10-02および時間22:59:10、右側にはAny Co.が表示されています。

chap4_db_nagivation_003.gifは、「Department」というタイトルのWebページを示しています。すぐ下に、ヘッダーとページの本体を区切る水平線があります。次に、2行4列の表があります。最初の行には、次の列ヘッダーが含まれています。

「Department ID」、「Department Name」、「Manager ID」、「Location ID」

データ行には、次のデータが含まれています。

「10」、「Administration」、「200」、「1700」

これらのデータの下には、「<Previous」および「Next>」というラベルの付いた2つのボタンがあります。

すぐ下に、ページの本体とフッターを区切る水平線があります。フッターの左側には日付2005-10-02および時間22:59:29、右側にはAny Co.が表示されています。

chap4_enhance_dept_001.gifは、「Department」というタイトルのWebページを示しています。すぐ下に、ヘッダーとページの本体を区切る水平線があります。次に、2行5列の表があります。最初の行には、次の列ヘッダーが含まれています。

「Department ID」、「Department Name」、「Number of Employees」、「Manager Name」、「Location」

データ行には、次のデータが含まれています。

「10」、「Administration」、「4」、「J. Whalen」、「United States of America」

これらのデータの下には、「<Previous」および「Next>」というラベルの付いた2つのボタンがあります。

すぐ下に、ページの本体とフッターを区切る水平線があります。フッターの左側には日付2005-10-03および時間10:56:55、右側にはAny Co.が表示されています。

chap4_basic_emp_001.gifは、「Employees」というタイトルのWebページを示しています。すぐ下に、ヘッダーとページの本体を区切る水平線があります。次に、9行5列の表があります。列ヘッダーは次のとおりです。

「Employee ID」、「Employee Name」、「Hiredate」、「Salary」、「Commission (%)」

データ行には、次のデータが含まれています。

「100」、「S. King」、「17-JUN-87」、「24,000.00」、「0」

「101」、「N. Kochhar」、「21-SEP-89」、「17,000.00」、「0」

「102」、「L. De Haan」、「13-JAN-93」、「17,000.00」、「0」

「103」、「A. Hunchold」、「03-JAN-90」、「9,000.00」、「0」

「104」、「B. Ernst」、「21-MAY-91」、「6,000.00」、「0」

「105」、「D. Austin」、「25-JUN-97」、「4,800.00」、「0」

「106」、「V. Pataballa」、「05-FEB-98」、「4,800.00」、「0」

「107」、「D. Lorentz」、「07-FEB-99」、「4,200.00」、「0」

「108」、「N. Greenberg」、「17-AUG-94」、「12,000.00」、「0」

chap5_test_emp_001.gifは、「Employees」というタイトルのWebページを示しています。すぐ下に、ヘッダーとページの本体を区切る水平線があります。次に、9行6列の表があります。列ヘッダーは次のとおりです。

ヘッダーなし、「Employee ID」、「Employee Name」、「Hiredate」、「Salary」、「Commission (%)」

データ行には、次のデータが含まれています。

選択解除されたラジオ・ボタン、「100」、「S. King」、「17-JUN-87」、「24,000.00」、「0」

選択解除されたラジオ・ボタン、「101」、「N. Kochhar」、「21-SEP-89」、「17,000.00」、「0」

選択解除されたラジオ・ボタン、「102」、「L. De Haan」、「13-JAN-93」、「17,000.00」、「0」

選択解除されたラジオ・ボタン、「103」、「A. Hunold」、「03-JAN-90」、「9,000.00」、「0」

選択解除されたラジオ・ボタン、「104」、「B. Ernst」、「21-MAY-91」、「6,000.00」、「0」

選択解除されたラジオ・ボタン、「105」、「D. Austin」、「25-JUN-97」、「4,800.00」、「0」

選択解除されたラジオ・ボタン、「106」、「V. Pataballa」、「05-FEB-98」、「4,800.00」、「0」

選択解除されたラジオ・ボタン、「107」、「D. Lorentz」、「07-FEB-99」、「4,200.00」、「0」

chap5_test_emp_002.gifは、Webページの最下部を示しています。6行6列の表があります。

データ行には、次のデータが含まれています。

選択解除されたラジオ・ボタン、「201」、「M. Hartstein」、「17-FEB-96」、「13,000.00」、「0」

選択解除されたラジオ・ボタン、「202」、「P. Fay」、「17-Aug-97」、「8,000.00」、「0」

選択解除されたラジオ・ボタン、「203」、「S. Mavris」、「07-JUN-94」、「6,500.00」、「0」

選択解除されたラジオ・ボタン、「204」、「H. Baer」、「07-JUN-94」、「10,000.00」、「0」

選択解除されたラジオ・ボタン、「205」、「S. Higgins」、「07-JUN-94」、「12,000.00」、「0」

選択解除されたラジオ・ボタン、「206」、「W. Gietz」、「07-JUN-94」、「8,300.00」、「0」

これらのデータの下には、「Modify」、「Delete」および「Insert new employee」というラベルの付いた3つのボタンがあります。

すぐ下に、ページの本体とフッターを区切る水平線があります。フッターの左側には日付2005-10-04および時間13:28:34、右側にはAny Co.が表示されています。

chap5_test_emp_003.gifは、Webページの最下部を示しています。1行6列の表があります。

データ行には、次のデータが含まれています。

選択解除されたラジオ・ボタン、「206」、「W. Gietz」、「07-JUN-94」、「8,300.00」、「0」

これらのデータの下には、「Modify」、「Delete」および「Insert new employee」というラベルの付いた3つのボタンがあります。マウス・カーソルは、「Insert new employee」ボタンを指しています。

すぐ下に、ページの本体とフッターを区切る水平線があります。フッターの左側には日付2005-10-04および時間13:28:34、右側にはAny Co.が表示されています。

chap5_test_emp_004.gifは、「Insert New Employee」というタイトルのWebフォームを示しています。すぐ下に、ヘッダーとフォームの本体を区切る水平線があります。次に、フォーム・ラベルおよび入力フィールドで構成されている7行2列の表があります。左側の列のフィールド・ラベルは次のとおりです。

「Department ID」、「First Name」、「Last Name」、「Hiredate」、「Job」、「Salary」、「Commission (%)」

2つ目の列の入力フィールドには、次の値が含まれています。

「10」、「James」、「Bond」、「04-OCT-05」、「Programmer」、「7000」、「0」

これらのフィールドの下には、「Save」および「Cancel」というラベルの付いた2つのボタンがあります。マウス・カーソルは、「Save」ボタンを指しています。

すぐ下に、フォームの本体とフッターを区切る水平線があります。フッターの左側には日付2005-10-04および時間13:31:27、右側にはAny Co.が表示されています。

chap5_test_emp_005.gifは、Webページの最下部を示しています。2行6列の表があります。

データ行には、次のデータが含まれています。

選択解除されたラジオ・ボタン、「206」、「W. Gietz」、「07-JUN-94」、「8,300.00」、「0」

選択解除されたラジオ・ボタン、「248」、「J. Bond」、「04-OCT-05」、「7,000.00」、「0」

これらのデータの下には、「Modify」、「Delete」および「Insert new employee」というラベルの付いた3つのボタンがあります。

すぐ下に、ページの本体とフッターを区切る水平線があります。フッターの左側には日付2005-10-04および時間13:40:42、右側にはAny Co.が表示されています。

chap5_test_emp_006は、Webページの最下部を示しています。2行6列の表があります。

データ行には、次のデータが含まれています。

選択解除されたラジオ・ボタン、「206」、「W. Gietz」、「07-JUN-94」、「8,300.00」、「0」

選択されたラジオ・ボタン、「248」、「J. Bond」、「04-OCT-05」、「7,000.00」、「0」

これらのデータの下には、「Modify」、「Delete」および「Insert new employee」というラベルの付いた3つのボタンがあります。マウス・カーソルは、「Modify」ボタンを指しています。

すぐ下に、ページの本体とフッターを区切る水平線があります。フッターの左側には日付2005-10-04および時間13:40:42、右側にはAny Co.が表示されています。

chap5_test_emp_007.gifは、「Modify Employee」というタイトルのWebフォームを示しています。すぐ下に、ヘッダーとフォームの本体を区切る水平線があります。次に、フォーム・ラベルおよび入力フィールドで構成されている6行2列の表があります。左側の列のフィールド・ラベルは次のとおりです。

「Employee ID」、「First Name」、「Last Name」、「Email Address」、「Salary」、「Commission (%)」

2つ目の列の最初の入力フィールドである「Employee ID」の値は、編集不可能で、248という値が入力されています。

2つ目の列の残りの入力フィールドには、次の値が入力されています。

「James」、「Bond」、「JBOND」、「7100」、「0」

これらのフィールドの下には、「Save」および「Cancel」というラベルの付いた2つのボタンがあります。マウス・カーソルは、「Save」ボタンを指しています。

すぐ下に、フォームの本体とフッターを区切る水平線があります。フッターの左側には日付2005-10-04および時間13:45:04、右側にはAny Co.が表示されています。

chap5_test_emp_008.gifは、Webページの最下部を示しています。1行6列の表があります。

データ行には、次のデータが含まれています。

選択解除されたラジオ・ボタン、「248」、「J. Bond」、「04-OCT-05」、「7,000.00」、「0」

これらのデータの下には、「Modify」、「Delete」および「Insert new employee」というラベルの付いた3つのボタンがあります。

すぐ下に、ページの本体とフッターを区切る水平線があります。フッターの左側には日付2005-10-04および時間13:47:38、右側にはAny Co.が表示されています。

chap5_test_emp_009.gifは、Webページの最下部を示しています。1行6列の表があります。

データ行には、次のデータが含まれています。

選択されたラジオ・ボタン、「248」、「J. Bond」、「04-OCT-05」、「7,000.00」、「0」

これらのデータの下には、「Modify」、「Delete」および「Insert new employee」というラベルの付いた3つのボタンがあります。マウス・カーソルは、「Delete」ボタンを指しています。

すぐ下に、ページの本体とフッターを区切る水平線があります。フッターの左側には日付2005-10-04および時間13:47:38、右側にはAny Co.が表示されています。

chap5_test_emp_010.gifは、Webページの最下部を示しています。1行6列の表があります。

データ行には、次のデータが含まれています。

選択解除されたラジオ・ボタン、「206」、「W. Gietz」、「07-JUN-94」、「8,300.00」、「0」

これらのデータの下には、「Modify」、「Delete」および「Insert new employee」というラベルの付いた3つのボタンがあります。

すぐ下に、ページの本体とフッターを区切る水平線があります。フッターの左側には日付2005-10-04および時間13:52:19、右側にはAny Co.が表示されています。

chap5_combine_deptemp_001.gifは、「Department: Administration」というタイトルのWebページを示しています。すぐ下に、ヘッダーとページの本体を区切る水平線があります。次に、2行5列の表があります。最初の行には、次の列ヘッダーが含まれています。

「Department ID」、「Department Name」、「Number of Employees」、「Manager Name」、「Location」

データ行には、次のデータが含まれています。

「10」、「Administration」、「1」、「J. Whalen」、「United States of America」

これらのデータの下には、「<Previous」、「Next>」および「Show Employees」というラベルの付いた3つのボタンがあります。

すぐ下に、ページの本体とフッターを区切る水平線があります。フッターの左側には日付2005-10-10および時間14:20:14、右側にはAny Co.が表示されています。

chap5_combine_deptemp_002.gifは、「Employees: Administration」というタイトルのWebページを示しています。すぐ下に、ヘッダーとページの本体を区切る水平線があります。次に、2行6列の表があります。最初の行には、次の列ヘッダーが含まれています。

ヘッダーなし、「Employee ID」、「Employee Name」、「Hiredate」、「Salary」、「Commission (%)」

データ行には、次のデータが含まれています。

選択解除されたラジオ・ボタン、「200」、「J. Whalen」、「17-Sep-87」、「4,400.00」、「0」

これらのデータの下には、「Modify」、「Delete」、「Insert new employee」および「Return to Departments」というラベルの付いた4つのボタンがあります。

すぐ下に、ページの本体とフッターを区切る水平線があります。フッターの左側には日付2005-10-10および時間14:24:45、右側にはAny Co.が表示されています。

chap5_err_handling_001.gifは、「Department: Administration」というタイトルのWebページを示しています。すぐ下に、ヘッダーとページの本体を区切る水平線があります。次に、2行5列の表があります。最初の行には、次の列ヘッダーが含まれています。

「Department ID」、「Department Name」、「Number of Employees」、「Manager Name」、「Location」

データ行には、次のデータが含まれています。

「10」、「Administration」、「1」、「J. Whalen」、「United States of America」

これらのデータの下には、「<Previous」、「Next>」および「Show Employees」というラベルの付いた3つのボタンがあります。マウス・カーソルは、「Next>」ボタンを指しています。

すぐ下に、ページの本体とフッターを区切る水平線があります。フッターの左側には日付2005-10-10および時間16:33:20、右側にはAny Co.が表示されています。

chap5_err_handling_002.gifは、「Department: Accounting」というタイトルのWebページを示しています。すぐ下に、ヘッダーとページの本体を区切る水平線があります。次に、2行5列の表があります。最初の行には、次の列ヘッダーが含まれています。

「Department ID」、「Department Name」、「Number of Employees」、「Manager Name」、「Location」

データ行には、次のデータが含まれています。

「110」、「Accounting」、「2」、「S. Higgins」、「United States of America」

これらのデータの下には、「<Previous」、「Next>」および「Show Employees」というラベルの付いた3つのボタンがあります。マウス・カーソルは、「Next>」ボタンを指しています。

すぐ下に、ページの本体とフッターを区切る水平線があります。フッターの左側には日付2005-10-10および時間16:34:07、右側にはAny Co.が表示されています。

chap5_err_handling_003.gifは、「Department: Administration」というタイトルのWebページを示しています。すぐ下に、ヘッダーとページの本体を区切る水平線があります。次に、2行5列の表があります。最初の行には、次の列ヘッダーが含まれています。

「Department ID」、「Department Name」、「Number of Employees」、「Manager Name」、「Location」

データ行には、次のデータが含まれています。

「10」、「Administration」、「1」、「J. Whalen」、「United States of America」

これらのデータの下には、「<Previous」、「Next>」および「Show Employees」というラベルの付いた3つのボタンがあります。マウス・カーソルは、「Show Employees」ボタンを指しています。

すぐ下に、ページの本体とフッターを区切る水平線があります。フッターの左側には日付2005-10-10および時間16:37:29、右側にはAny Co.が表示されています。

chap5_err_handling_004.gifは、「Employees: Administration」というタイトルのWebページを示しています。すぐ下に、ヘッダーとページの本体を区切る水平線があります。次に、2行6列の表があります。最初の行には、次の列ヘッダーが含まれています。

ヘッダーなし、「Employee ID」、「Employee Name」、「Hiredate」、「Salary」、「Commission (%)」

データ行には、次のデータが含まれています。

選択解除されたラジオ・ボタン、「200」、「J. Whalen」、「17-Sep-87」、「4,400.00」、「0」

これらのデータの下には、「Modify」、「Delete」、「Insert new employee」および「Return to Departments」というラベルの付いた4つのボタンがあります。マウス・カーソルは、「Insert new employee」ボタンを指しています。

すぐ下に、ページの本体とフッターを区切る水平線があります。フッターの左側には日付2005-10-10および時間16:37:54、右側にはAny Co.が表示されています。

chap5_err_handling_005.gifは、「Insert New Employee」というタイトルのWebフォームを示しています。すぐ下に、ヘッダーとフォームの本体を区切る水平線があります。次に、フォーム・ラベルおよび入力フィールドで構成されている7行2列の表があります。左側の列のフィールド・ラベルは次のとおりです。

「Department ID」、「First Name」、「Last Name」、「Hiredate」、「Job」、「Salary」、「Commission (%)」

2つ目の列の入力フィールドには、次の値が含まれています。

「99」、「New」、「Person」、「10-OCT-05」、「Accountant」、「1000」、「0」

これらのフィールドの下には、「Save」および「Cancel」というラベルの付いた2つのボタンがあります。マウス・カーソルは、「Save」ボタンを指しています。

すぐ下に、フォームの本体とフッターを区切る水平線があります。フッターの左側には日付2005-10-10および時間16:38:06、右側にはAny Co.が表示されています。

chap5_err_handling_006.gifは、「Department 99 does not yet exist」というタイトルのWebページを示しています。すぐ下に、ヘッダーとページの本体を区切る水平線があります。その下に、次のエラー・メッセージがあります。

```
Error at line 86 of /home/gstokol/public_html/chap5/anyco_db.inc
```

```
ORA-02291: Integrity constraint (HR.EMP_DEPT_FK) violated - parent key not found
```

エラー・メッセージの下には、「Return to Departments」というラベルの付いたボタンがあります。マウス・カーソルは、「Return to Departments」ボタンを指しています。

すぐ下に、ページの本体とフッターを区切る水平線があります。フッターの左側には日付2005-10-10および時間16:39:15、右側にはAny Co.が表示されています。

chap6_hrgrantproc.gifは、SQL DeveloperでのSQLワークシートを示しています。「Enter SQL Statement」ペインには、文が含まれています。

```
grant create procedure to hr;
```

「Script Output」ペインには、メッセージが含まれています。

```
grant create succeeded.
```

chap6_hrcalcrem.gifは、SQL DeveloperでのSQLワークシートを示しています。

「Enter SQL statement」 ペインには、文が含まれています。

```
create or replace function calc_remuneration(  
    salary IN number, commission_pct IN number) return number is  
begin  
    return ((salary * 12) + (salary * 12 * nvl(commission_pct,0)));  
end;
```

「Script Output」 ペインには、メッセージが含まれています。

```
function calc_remuneration( Compiled.
```

chap6_stored_proc_test_001.gifは、「Department: Administration」というタイトルのWebページを示しています。すぐ下に、ヘッダーとページの本体を区切る水平線があります。次に、2行5列の表があります。最初の行には、次の列ヘッダーが含まれています。

「Department ID」、「Department Name」、「Number of Employees」、「Manager Name」、「Location」

データ行には、次のデータが含まれています。

「10」、「Administration」、「1」、「J. Whalen」、「United States of America」

これらのデータの下には、「Previous」、「Next」および「Show Employees」というラベルの付いた3つのボタンがあります。

すぐ下に、ページの本体とフッターを区切る水平線があります。フッターの左側には日付2005-10-10および時間22:12:54、右側にはAny Co.が表示されています。

chap6_stored_proc_test_002.gifは、「Employees: Administration」というタイトルのWebページを示しています。すぐ下に、ヘッダーとページの本体を区切る水平線があります。次に、2行7列の表があります。最初の行には、次の列ヘッダーが含まれています。

ヘッダーなし、「Employee ID」、「Employee Name」、「Hiredate」、「Salary」、「Commission (%)」、「Remuneration」

データ行には、次のデータが含まれています。

選択解除されたラジオ・ボタン、「200」、「J. Whalen」、「17-Sep-87」、「4,400.00」、「0」、「52,800.00」

これらのデータの下には、「Modify」、「Delete」、「Insert new employee」および「Return to Departments」というラベルの付いた4つのボタンがあります。

すぐ下に、ページの本体とフッターを区切る水平線があります。フッターの左側には日付2005-10-10および時間22:14:31、右側にはAny Co.が表示されています。

chap6_hrcreatepack.gifは、SQL DeveloperでのSQLワークシートを示しています。

「Enter SQL statement」 ペインには、文が含まれています。

```
CREATE OR REPLACE PACKAGE cv_types AS
  TYPE empinfotyp IS REF CURSOR;
  PROCEDURE get_employees(deptid in number,
                          employees in out empinfotyp);
END cv_types;
```

「Script Output」 ペインには、メッセージが含まれています。

```
Package cv_types Compiled.
```

cchap6_hrcreatebody.gifは、SQL DeveloperでのSQLワークシートを示しています。

「Enter SQL statement」 ペインには、文が含まれています。

```
CREATE OR REPLACE PACKAGE BODY cv_types AS
  PROCEDURE get_employees(deptid in number,
                          employees in out empinfotyp)
  IS
  BEGIN
    OPEN employees FOR
    SELECT employee_id,
           substr(first_name,1,1) || '. ' || last_name as employee_name,
           hire_date,
           to_char(salary, '999G999D99') as salary,
```

「Script Output」 ペインには、メッセージが含まれています。

Package Body Compiled.

chap6_refcursor_005.gifは、「Department: Administration」というタイトルのWebページを示しています。すぐ下に、ヘッダーとページの本体を区切る水平線があります。次に、2行5列の表があります。最初の行には、次の列ヘッダーが含まれています。

「Department ID」、「Department Name」、「Number of Employees」、「Manager Name」、「Location」

データ行には、次のデータが含まれています。

「10」、「Administration」、「1」、「J. Whalen」、「United States of America」

これらのデータの下には、「Previous」、「Next」および「Show Employees」というラベルの付いた3つのボタンがあります。

すぐ下に、ページの本体とフッターを区切る水平線があります。フッターの左側には日付2005-10-10および時間23:27:47、右側にはAny Co.が表示されています。

chap6_refcursor_006.gifは、「Department: Marketing」というタイトルのWebページを示しています。すぐ下に、ヘッダーとページの本体を区切る水平線があります。次に、2行5列の表があります。最初の行には、次の列ヘッダーが含まれています。

「Department ID」、「Department Name」、「Number of Employees」、「Manager Name」、「Location」

データ行には、次のデータが含まれています。

「20」、「Marketing」、「2」、「M. Hartstein」、「Canada」

これらのデータの下には、「Previous」、「Next」および「Show Employees」というラベルの付いた3つのボタンがあります。

すぐ下に、ページの本体とフッターを区切る水平線があります。フッターの左側には日付2005-10-10および時間23:28:36、右側にはAny Co.が表示されています。

chap6_refcursor_007.gifは、「Employees: Marketing」というタイトルのWebページを示しています。すぐ下に、ヘッダーとページの本体を区切る水平線があります。次に、3行7列の表があります。最初の行には、次の列ヘッダーが含まれています。

ヘッダーなし、「Employee ID」、「Employee Name」、「Hiredate」、「Salary」、「Commission (%)」、「Remuneration」

データ行には、次のデータが含まれています。

選択解除されたラジオ・ボタン、「201」、「M. Hartstein」、「17-Feb-96」、「13,000.00」、「0」、「156,000.00」

選択解除されたラジオ・ボタン、「202」、「P. Fay」、「17-Aug-97」、「6,000.00」、「0」、「72,000.00」

これらのデータの下には、「Modify」、「Delete」、「Insert new employee」および「Return to Departments」というラベルの付いた4つのボタンがあります。

すぐ下に、ページの本体とフッターを区切る水平線があります。フッターの左側には日付2005-10-10および時間23:30:32、右側にはAny Co.が表示されています。

chap7_createtable.gifは、SQL DeveloperでのSQLワークシートを示しています。

「Enter SQL Statement」 ペインには、文が含まれています。

```
CREATE TABLE employee_photos(  
  employee_id      NUMBER,  
  employee_thumbnail BLOB);
```

「Script Output」 ペインには、メッセージが含まれています。

```
CREATE TABLE Succeeded.
```

chap7_loading_005.gifは、「Department: Administration」というタイトルのWebページを示しています。すぐ下に、ヘッダーとページの本体を区切る水平線があります。次に、2行5列の表があります。最初の行には、次の列ヘッダーが含まれています。

「Department ID」、「Department Name」、「Number of Employees」、「Manager Name」、「Location」

データ行には、次のデータが含まれています。

「10」、「Administration」、「1」、「J. Whalen」、「United States of America」

これらのデータの下には、「Previous」、「Next」および「Show Employees」というラベルの付いた3つのボタンがあります。

すぐ下に、ページの本体とフッターを区切る水平線があります。フッターの左側には日付2005-10-11および時間11:18:29、右側にはAny Co.が表示されています。

chap7_loading_006.gifは、「Employees: Administration」というタイトルのWebページを示しています。すぐ下に、ヘッダーとページの本体を区切る水平線があります。次に、2行8列の表があります。最初の行には、次の列ヘッダーが含まれています。

ヘッダーなし、「Employee ID」、「Employee Name」、「Hiredate」、「Salary」、「Commission (%)」、「Remuneration」、「Photograph」

データ行には、次のデータが含まれています。

選択解除されたラジオ・ボタン、「200」、「J. Whalen」、「17-Sep-87」、「4,400.00」、「0」、「52,800.00」、「Employee photo」(テキスト)

これらのデータの下には、「Modify」、「Delete」、「Insert new employee」および「Return to Departments」というラベルの付いた4つのボタンがあります。

すぐ下に、ページの本体とフッターを区切る水平線があります。フッターの左側には日付2005-10-11および時間11:19:31、右側にはAny Co.が表示されています。

chap7_loading_007.gifは、「Insert New Employee」というタイトルのWebフォームを示しています。すぐ下に、ヘッダーとフォームの本体を区切る水平線があります。次に、フォーム・ラベルおよび入力フィールドで構成されている8行2列の表があります。左側の列のフィールド・ラベルは次のとおりです。

「Department ID」、「First Name」、「Last Name」、「Hiredate」、「Job」、「Salary」、「Commission (%)」、「Photo」

2つ目の列の入力フィールドには、次の値が含まれています。

「10」、「Glenn」、「Stoko1」、「11-OCT-05」、「Programmer」、「8000」、「0」、NULL

「Photo」入力フィールドの右側に、「Browse」ボタンがあります。マウス・カーソルは、「Browse」ボタンを指しています。

これらのフィールドの下には、「Save」および「Cancel」というラベルの付いた2つのボタンがあります。

すぐ下に、フォームの本体とフッターを区切る水平線があります。フッターの左側には日付2005-10-11および時間11:21:05、右側にはAny Co.が表示されています。

chap7_loading_008.gifは、「File Upload」ダイアログ・ボックスを示しています。右側に、public_htmlディレクトリのコンテンツを表示していることが示されています。ファイルnewemp.jpgが選択されています。マウス・カーソルは、「Open」ボタンを指しています。

chap7_loading_022.gifは、「Insert New Employee」というタイトルのWebフォームを示しています。すぐ下に、ヘッダーとフォームの本体を区切る水平線があります。次に、フォーム・ラベルおよび入力フィールドで構成されている8行2列の表があります。左側の列のフィールド・ラベルは次のとおりです。

「Department ID」、「First Name」、「Last Name」、「Hiredate」、「Job」、「Salary」、「Commission (%)」、「Photo」

2つ目の列の入力フィールドには、次の値が含まれています。

「10」、「Chris」、「Jones」、「11-OCT-05」、「Marketing Manager」、「9000」、「0」、NULL

「Photo」入力フィールドの右側に、「Browse」ボタンがあります。

これらのフィールドの下には、「Save」および「Cancel」というラベルの付いた2つのボタンがあります。マウス・カーソルは、「Save」ボタンを指しています。

すぐ下に、フォームの本体とフッターを区切る水平線があります。フッターの左側には日付2005-10-11および時間12:32:04、右側にはAny Co.が表示されています。

chap7_loading_009.gifは、「Employees: Administration」というタイトルのWebページを示しています。すぐ下に、ヘッダーとページの本体を区切る水平線があります。次に、3行8列の表があります。最初の行には、次の列ヘッダーが含まれています。

ヘッダーなし、「Employee ID」、「Employee Name」、「Hiredate」、「Salary」、「Commission (%)」、「Remuneration」、「Photograph」

データ行には、次のデータが含まれています。

選択解除されたラジオ・ボタン、「200」、「J. Whalen」、「17-Sep-87」、「4,400.00」、「0」、「52,800.00」、「Employee photo」(テキスト)

選択解除されたラジオ・ボタン、「209」、「G. Stokol」、「11-Sep-05」、「8,000.00」、「0」、「96,000.00」、「Employee photo」(イメージ)

これらのデータの下には、「Modify」、「Delete」、「Insert new employee」および「Return to Departments」というラベルの付いた4つのボタンがあります。

すぐ下に、ページの本体とフッターを区切る水平線があります。フッターの左側には日付2005-10-11および時間12:27:16、右側にはAny Co.が表示されています。

chap7_loading_018.gifは、「Department: Administration」というタイトルのWebページを示しています。すぐ下に、ヘッダーとページの本体を区切る水平線があります。次に、2行5列の表があります。最初の行には、次の列ヘッダーが含まれています。

「Department ID」、「Department Name」、「Number of Employees」、「Manager Name」、「Location」

データ行には、次のデータが含まれています。

「10」、「Administration」、「2」、「J. Whalen」、「United States of America」

これらのデータの下には、「Previous」、「Next」および「Show Employees」というラベルの付いた3つのボタンがあります。マウス・カーソルは、「Show Employees」ボタンを指しています。

すぐ下に、ページの本体とフッターを区切る水平線があります。フッターの左側には日付2005-10-11および時間12:29:43、右側にはAny Co.が表示されています。

chap7_loading_019.gifは、「Employees: Administration」というタイトルのWebページを示しています。すぐ下に、ヘッダーとページの本体を区切る水平線があります。次に、3行7列の表があります。最初の行には、次の列ヘッダーが含まれています。

ヘッダーなし、「Employee ID」、「Employee Name」、「Hiredate」、「Salary」、「Commission (%)」、「Remuneration」

データ行には、次のデータが含まれています。

選択解除されたラジオ・ボタン、「200」、「J. Whalen」、「17-Sep-87」、「4,400.00」、「0」、「52,800.00」、「Employee photo」(テキスト)

選択解除されたラジオ・ボタン、「209」、「G. Stokol」、「11-Sep-05」、「8,000.00」、「0」、「96,000.00」

これらのデータの下には、「Modify」、「Delete」、「Insert new employee」および「Return to Departments」というラベルの付いた4つのボタンがあります。マウス・カーソルは、「Insert new employee」ボタンを指しています。

すぐ下に、ページの本体とフッターを区切る水平線があります。フッターの左側には日付2005-10-11および時間12:30:46、右側にはAny Co.が表示されています。

chap7_loading_020.gifは、「Insert New Employee」というタイトルのWebフォームを示しています。すぐ下に、ヘッダーとフォームの本体を区切る水平線があります。次に、フォーム・ラベルおよび入力フィールドで構成されている8行2列の表があります。左側の列のフィールド・ラベルは次のとおりです。

「Department ID」、「First Name」、「Last Name」、「Hiredate」、「Job」、「Salary」、「Commission (%)」、「Photo」

2つ目の列の入力フィールドには、次の値が含まれています。

「10」、「Chris」、「Jones」、「11-OCT-05」、「Marketing Manager」、「9000」、「0」、NULL

「Photo」入力フィールドの右側に、「Browse」ボタンがあります。マウス・カーソルは、「Browse」ボタンを指しています。

これらのフィールドの下には、「Save」および「Cancel」というラベルの付いた2つのボタンがあります。

すぐ下に、フォームの本体とフッターを区切る水平線があります。フッターの左側には日付2005-10-11および時間12:32:04、右側にはAny Co.が表示されています。

chap7_loading_021.gifは、「File Upload」ダイアログ・ボックスを示しています。右側に、public_htmlディレクトリのコンテンツを表示していることが示されています。ファイルnewemp.jpgが選択されています。マウス・カーソルは、「Open」ボタンを指しています。

chap7_loading_023.gifは、「Employees: Administration」というタイトルのWebページを示しています。すぐ下に、ヘッダーとページの本体を区切る水平線があります。次に、4行8列の表があります。最初の行には、次の列ヘッダーが含まれています。

ヘッダーなし、「Employee ID」、「Employee Name」、「Hiredate」、「Salary」、「Commission (%)」、「Remuneration」、「Photograph」

データ行には、次のデータが含まれています。

選択解除されたラジオ・ボタン、「200」、「J. Whalen」、「17-Sep-87」、「4,400.00」、「0」、「52,800.00」、「Employee photo」(テキスト)

選択解除されたラジオ・ボタン、「209」、「G. Stokol」、「11-Sep-05」、「8,000.00」、「0」、「96,000.00」、「Employee photo」(イメージ)

選択解除されたラジオ・ボタン、「210」、「C. Jones」、「11-Oct-05」、「9,000.00」、「0」、「108,000.00」、「Employee photo」(サムネイル・イメージ)

これらのデータの下には、「Modify」、「Delete」、「Insert new employee」および「Return to Departments」というラベルの付いた4つのボタンがあります。