

ご注意

本書および本書で紹介する製品をご使用になる前に、53ページの『付録 B. 特記事項』に記載されている情報をお読みください。

当版に関する特記事項

本書には、IBM の専有情報が含まれています。その情報は、使用許諾条件に基づき提供され、著作権により保護されています。本書に記載される情報には、いかなる製品の保証も含まれていません。また、本書で提供されるいかなる記述も、製品保証として解釈すべきではありません。

IBM 資料は、オンラインでご注文いただくことも、ご自分の国または地域の IBM 担当員を通してお求めいただくこともできます。

- オンラインで資料を注文するには、www.ibm.com/shop/publications/order にある IBM Publications Center をご利用ください。
- ご自分の国または地域の IBM 担当員を見つけるには、www.ibm.com/planetwide にある IBM Directory of Worldwide Contacts をお調べください。

IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。

お客様の環境によっては、資料中の円記号がバックスラッシュと表示されたり、バックスラッシュが円記号と表示されたりする場合があります。

原 典： SC23-5854-00
DB2 Version 9.5 for Linux, UNIX, and Windows
Developing Perl and PHP Applications

発 行： 日本アイ・ビー・エム株式会社

担 当： ナショナル・ランゲージ・サポート

目次

第 1 部 PHP アプリケーションの開発 1

第 1 章 DB2 用の PHP アプリケーション開発の概要 3

第 2 章 PHP 環境のセットアップ 5

Windows での PHP 環境のセットアップ 5

Linux または UNIX での PHP 環境のセットアップ . 6

第 3 章 ibm_db2 を使用した開発 9

PHP による DB2 データベースへの接続 (ibm_db2) . 9

データベース・メタデータの取得 (ibm_db2) 10

PHP での XQuery 式の実行 (ibm_db2) 11

SQL ステートメントの実行 12

PHP での単一 SQL ステートメントの実行 (ibm_db2) 12

PHP での SQL ステートメントの準備と実行 (ibm_db2) 13

PHP でのラージ・オブジェクトの挿入 (ibm_db2) 15

照会結果セットの読み取り 16

PHP での結果セットからの列のフェッチ (ibm_db2) 16

PHP での結果セットからの行の取り出し (ibm_db2) 17

PHP でのラージ・オブジェクトのフェッチ (ibm_db2) 17

PHP でのトランザクションの管理 (ibm_db2) 18

エラーおよび警告メッセージの処理 (ibm_db2). . . 19

ストアード・プロシージャの呼び出し 20

PHP での、OUT または INOUT パラメーターを持つストアード・プロシージャの呼び出し (ibm_db2) 20

PHP での、複数の結果セットを戻すストアード・プロシージャの呼び出し (ibm_db2). 21

第 4 章 PDO を使用した開発 23

PHP による DB2 データベースへの接続 (PDO) . . . 23

SQL ステートメントの実行 24

PHP での、結果セットを戻さない単一 SQL ステートメントの実行 (PDO) 24

PHP での、結果セットを戻す単一 SQL ステートメントの実行 (PDO) 25

SQL ステートメントの準備と実行 (PDO) 25

PHP でのラージ・オブジェクトの挿入 (PDO) . . . 27

照会結果セットの読み取り 28

PHP での結果セットからの列のフェッチ (PDO) 28

PHP での結果セットからの行の取り出し (PDO) 28

PHP でのラージ・オブジェクトのフェッチ (PDO) 30

PHP でのトランザクションの管理 (PDO) 31

PHP でのエラーおよび警告の処理 (PDO) 32

ストアード・プロシージャの呼び出し 32

PHP での、OUT または INOUT パラメーターを持つストアード・プロシージャの呼び出し (PDO) 32

PHP での、複数の結果セットを戻すストアード・プロシージャの呼び出し (PDO) 34

第 2 部 Perl アプリケーションの開発 35

第 5 章 Perl でのプログラミングに関する考慮事項 37

Perl でのデータベース接続 37

Perl での取り出し結果 38

Perl のパラメーター・マーカー 38

Perl の SQLSTATE および SQLCODE 変数 39

Perl の制約事項 39

Perl プログラムの例 39

Perl アプリケーションの構築 40

第 3 部 付録 41

付録 A. DB2 技術情報の概説 43

DB2 テクニカル・ライブラリー (ハードコピーまたは PDF 形式). 44

DB2 の印刷資料の注文方法 46

コマンド行プロセッサから SQL 状態ヘルプを表示する 47

異なるバージョンの DB2 インフォメーション・センターへのアクセス 47

DB2 インフォメーション・センターにおける特定の言語でのトピックの表示 48

コンピューターまたはイントラネット・サーバーにインストールされた DB2 インフォメーション・センターの更新 49

DB2 チュートリアル 51

DB2 トラブルシューティング情報 51

ご利用条件 52

付録 B. 特記事項 53

索引 57

第 1 部 PHP アプリケーションの開発

第 1 章 DB2 用の PHP アプリケーション開発の概要

PHP: Hypertext Preprocessor (PHP) とは、主に Web アプリケーションの開発を対象とした解釈済みプログラミング言語です。PHP の最初のバージョンは、1995 年に、Rasmus Lerdorf によって作成され、オープン・ソース・ライセンスのもとに提供されました。PHP は最初は非常に単純な HTML テンプレート・エンジンでしたが、時の経過とともに、PHP の開発者はデータベース・アクセス機能を追加し、インタープリターを再作成し、オブジェクト指向のサポートを導入し、パフォーマンスを向上させました。今日では、実用的なソリューションと、Web アプリケーションで最も一般に必要とされる機能のサポートに焦点を合わせているため、PHP は Web アプリケーション開発用に人気のある言語となりました。

Linux[®]、UNIX[®]、または Windows[®] オペレーティング・システム上で最も簡単なインストールおよび構成を行う方法として、実動システムで使用するための Zend Core for IBM をダウンロードしてインストールすることができます。Zend Core for IBM の有料サポートは、Zend から入手できます。Windows 上では、プリコンパイルされたバイナリー・バージョンの PHP が、<http://php.net/> からダウンロードできます。ほとんどの Linux ディストリビューションには、プリコンパイル・バージョンの PHP が組み込まれています。プリコンパイル・バージョンの PHP が組み込まれていない UNIX オペレーティング・システム上では、ユーザー独自のバージョンの PHP をコンパイルできます。

PHP はモジュラー言語であり、拡張モジュールを使用することによって、使用できる機能をカスタマイズできます。これらの拡張モジュールを使用すれば、XML の読み取り、書き込み、および操作、SOAP クライアント/サーバーの作成、およびサーバーとブラウザーとの間の通信の暗号化などのタスクを単純化できます。ただし、PHP の最も一般的な拡張モジュールは、データベースへの読み取り/書き込みアクセスを提供するものであり、これにより動的なデータベース・ドリブン Web サイトを簡単に作成できます。

PHP アプリケーション・オブジェクト (PDO) インターフェースのユーザー用に `pdo_ibm` という新規拡張モジュールを開発することにより、既存の PHP サポートをさらに発展させました。この新規拡張モジュールは、既存の `ibm_db2` 拡張モジュールと共に IBM Data Server Client の一部として組み込まれ、さらに便利になりました。`ibm_db2` および `pdo_ibm` の最新バージョンは、PHP Extension Community Library (PECL) <http://pecl.php.net/> から入手できます。PHP アプリケーションを通して DB2[®] データベースに保管されたデータにアクセスするには、どちらの拡張モジュールも使用できます。これらの拡張モジュール間の違いは次のとおりです。

- `ibm_db2` は、IBM[®] により DB2 データベースへのアクセス用に作成、保守、およびサポートされる拡張モジュールです。`ibm_db2` 拡張モジュールは、プロシージャ型アプリケーション・プログラミング・インターフェース (API) を提供します。これは通常のデータベースの作成、読み取り、更新、および書き込み操作に加え、データベース・メタデータへの広範なアクセスも提供します。`ibm_db2` 拡張モジュールは、PHP 4 または PHP 5 のいずれかでコンパイルできます。

- pdo_ibm は、PDO (PHP Data Objects) 拡張モジュール用のドライバーであり、PHP 5.1 で導入された標準オブジェクト指向データベース・インターフェースによる DB2 データベースへのアクセスを提供します。

3 番目の拡張モジュールである Unified ODBC は、これまで DB2 データベース・システムへのアクセスを提供してきました。ibm_db2 および pdo_ibm の両方には、Unified ODBC を上回るパフォーマンスおよび安定度における大きな利点があるため、新しいアプリケーションを Unified ODBC 拡張モジュールを使用して作成することはお勧めしません。ibm_db2 拡張モジュール API により、Unified ODBC 用に以前に作成されたアプリケーションの移植は、ほぼ、アプリケーションのソース・コード全体でグローバルに odbc_ 関数名を db2_ に変更するだけで容易に行うことができます。

第 2 章 PHP 環境のセットアップ

Windows での PHP 環境のセットアップ

DB2 は、`ibm_db2` 拡張モジュールまたは `pdo_ibm driver for the PHP Data Objects (PDO)` 拡張モジュール (あるいはその両方) を使用することにより、PHP プログラミング言語で作成されたクライアント・アプリケーションのデータベース・アクセスをサポートしています。Windows 上で DB2 をサポートしている PHP のバイナリー・バージョンをインストールするには、無償で入手可能な Zend Core for IBM を <http://zend.com/core/ibm/> からダウンロードしてインストールできます。あるいは、PHP のプリコンパイルされたバイナリー・バージョンを Windows に手動でインストールすることもできます。

Apache HTTP Server がシステムにインストールされている必要があります。

Windows に PHP のプリコンパイルされたバージョンを <http://www.php.net> からインストールして、サポートを使用可能にするには、以下のようになります。

1. <http://www.php.net> から PHP ZIP パッケージの最新バージョンと PECL モジュールの ZIP パッケージのコレクションをダウンロードします。本書の執筆時点における PHP の最新バージョンは PHP 5.2.3 です。
2. PHP ZIP パッケージをインストール・ディレクトリーに解凍します。
3. PECL モジュール ZIP パッケージのコレクションを、PHP インストール・ディレクトリーの `\ext\` サブディレクトリーに解凍します。
4. `php.ini-recommended` ファイルのコピーを作成して、インストール・ディレクトリーに `php.ini` という名前の新しいファイルを作成します。
5. テキスト・エディターで `php.ini` ファイルを開き、以下の行を追加します。
 - PDO 拡張モジュールと `pdo_ibm` ドライバーを使用可能にするには、以下のようになります。

```
extension=php_pdo.dll
extension=php_pdo_ibm.dll
```

- `ibm_db2` 拡張モジュールを使用可能にするには、以下のようになります。

```
extension=php_ibm_db2.dll
```

6. 以下の行を `httpd.conf` ファイルに追加して、Apache HTTP Server 2.x での PHP サポートを使用可能にします。ここで、`phpdir` は PHP インストール・ディレクトリーのことを指します。

```
LoadModule php5_module 'phpdir/php5apache2.dll'
AddType application/x-httpd-php .php
PHPIniDir 'phpdir'
```

7. Apache HTTP Server を再始動して、変更した構成を有効にします。

Linux または UNIX での PHP 環境のセットアップ

DB2 は、`ibm_db2` 拡張モジュールまたは `pdo_ibm` driver for the PHP Data Objects (PDO) 拡張モジュール (あるいはその両方) を使用することにより、PHP プログラミング言語で作成されたクライアント・アプリケーションのデータベース・アクセスをサポートしています。Linux または AIX 上で DB2 をサポートしている PHP のバイナリー・バージョンをインストールするには、無償で入手可能な Zend Core for IBM を <http://zend.com/core/ibm/> からダウンロードしてインストールできます。あるいは、手動でソースから PHP をコンパイルしてインストールすることもできます。

- Apache HTTP Server がシステムにインストールされている必要があります。
- DB2 開発ヘッダー・ファイルおよびライブラリーがシステムにインストールされている必要があります。
- gcc コンパイラーおよび他の開発パッケージ (`apache-devel`、`autoconf`、`automake`、`bison`、`flex`、`gcc`、および `libxml2-devel` パッケージを含む) がシステムにインストールされている必要があります。

Linux または UNIX でサポートする PHP をソースからコンパイルするには、以下のようになります。

1. <http://www.php.net> から PHP tarball の最新バージョンをダウンロードします。本書の執筆時点における PHP の最新バージョンは PHP 5.2.3 です。
2. 以下のコマンドを発行して、ファイルを `untar` します。

```
tar -xjf php-5.x.x.tar.bz2
```

3. 新しく作成された `php-5.x.x` ディレクトリーに移動します。
4. `configure` コマンドを発行して、`Makefile` を構成します。カスタム・バージョンの PHP に含める機能と拡張モジュールを指定してください。標準的な `configure` コマンドには、以下のオプションが含まれます。

```
./configure --enable-cli --disable-cgi --with-apxs2=/usr/sbin/apxs2  
--with-zlib --with-pdo-ibm=<sqlib>
```

`configure` のオプションには、以下の効果があります。

--enable-cli

PHP アクセスのコマンド行モードを使用可能にします。

--disable-cgi

PHP アクセスの CGI (Common Gateway Interface) モードを使用不可にします。

--with-apxs2=/usr/sbin/apxs2

PHP アクセスの Apache 2 動的共有オブジェクト (DSO) モードを使用可能にします。

--with-zlib

`zlib` 圧縮サポートを使用可能にします。

--with-pdo-ibm=<sqlib>

DB2 CLI・ライブラリーを使用する `pdo_ibm` ドライバーがデータベース・システムにアクセスできるようにします。`<sqlib>` 設定は、DB2 がインストールされているディレクトリーを指します。

5. `make` コマンドを発行して、ファイルをコンパイルします。
6. `make install` コマンドを発行して、ファイルをインストールします。
`configure` コマンドを使用して PHP インストール・ディレクトリーを構成した方法によっては、このコマンドを正常に実行するために `root` 権限が必要となる場合があります。これにより、実行可能ファイルがインストールされ、PHP をサポートするように Apache HTTP Server 構成が更新されます。
7. `root` 権限を持つユーザーとして以下のコマンドを発行して、`ibm_db2` 拡張モジュールをインストールします。

```
pecl install ibm_db2
```

このコマンドにより、PHP 用の `ibm_db2` 拡張モジュールのダウンロード、構成、コンパイル、およびインストールが行われます。

8. `php.ini-recommended` ファイルを、新しい PHP インストールの構成ファイル・パスにコピーします。構成ファイル・パスを判別するには、`php -i` コマンドを発行して、`php.ini` キーワードを探します。ファイルを `php.ini` に名前変更します。
9. 新しい `php.ini` ファイルをテキスト・エディターで開き、以下の行を追加します。`instance` は Linux または UNIX 上の DB2 インスタンスの名前を指しています。
 - `pdo_ibm` 用の DB2 環境を設定するには、以下のようになります。

```
PDO_IBM.db2_instance_name=instance
```
 - (Linux または UNIX) `ibm_db2` 拡張モジュールを使用可能にして、DB2 環境を設定するには、次のようになります。

```
extension=ibm_db2.so  
ibm_db2.instance_name=instance
```
10. Apache HTTP Server を再始動して、変更した構成を有効にします。

第 3 章 ibm_db2 を使用した開発

PHP による DB2 データベースへの接続 (ibm_db2)

DB2 データ・ソースに対してデータの作成、更新、削除、または検索を行うには、あらかじめ DB2 データベースに接続する必要があります。PHP 用の `ibm_db2` 拡張モジュールにより、DB2 データベース管理システムへのカタログ式接続または直接 TCP/IP 接続を使用して DB2 データベースに接続することができます。また、持続的なデータベース接続を行うこともできます。持続的なデータベース接続では、PHP からデータベースへの接続をオープンしたままにしておき、同一セットの ID とパスワードを持つ PHP スクリプトが接続を要求したときにその接続を再利用するという方法によって、パフォーマンスを向上させます。

`ibm_db2` 拡張モジュールを介して DB2 データベースに接続する前に、システム上に PHP 環境をセットアップし、`ibm_db2` 拡張モジュールを使用可能にする必要があります。

1. DB2 データベースへの接続を作成します。
 - DB2 データベースへの持続的でないデータベース接続を作成するには、カタログしたデータベース名または TCP/IP 直接接続に使用する完全なデータベース接続ストリングのどちらかを `database` 値として `db2_connect()` に指定して呼び出します。
 - DB2 データベースへの持続的なデータベース接続を作成するには、カタログしたデータベース名または TCP/IP 直接接続に使用する完全なデータベース接続ストリングのどちらかを `database` 値として `db2_pconnect()` に指定して呼び出します。
2. `db2_connect()` または `db2_pconnect` によって戻される値を確認します。
 - `db2_connect()` または `db2_pconnect` によって戻される値が `FALSE` である場合、接続に失敗しています。 `db2_conn_error()` および `db2_conn_errormsg()` によって診断情報を取得できます。
 - `db2_connect()` または `db2_pconnect` によって戻される値が `FALSE` でない場合、接続に成功しています。その接続リソースを使用して他の `ibm_db2` 関数によりデータの作成、更新、削除、または検索を行うことができます。

`db2_connect()` を呼び出して接続を作成した場合、以下の操作を行ったときに、PHP はデータベースへの接続を閉じます。

- 接続に対して `db2_close()` を呼び出す
- 接続リソースを `NULL` に設定する
- または、PHP スクリプトが終了する

`db2_pconnect()` を呼び出して接続を作成する場合、PHP は指定された接続リソースに対する `db2_close()` の呼び出しをすべて無視し、それ以後の PHP スクリプトに対してデータベースへの接続を常に開いたままにします。

データベース・メタデータの取得 (ibm_db2)

アプリケーションのクラス (管理インターフェースなど) によっては、任意のデータベースに含まれる構造および SQL オブジェクトを動的に反映する必要があるものがあります。データベースについてのメタデータを取得する 1 つの方法は、システム・カタログ表に対して SELECT ステートメントを直接発行することです。しかし、システム・カタログ表のスキーマは DB2 のバージョン間で異なることがあります。あるいは、DB2 Database for Linux, UNIX, and Windows 上のシステム・カタログ表のスキーマが DB2 for z/OS[®] 上のシステム・カタログ表のスキーマと異なることがあります。PHP 用の ibm_db2 拡張モジュールでは、これらの相違をアプリケーション・コードで記述する必要はなく、DB2 Database for Linux, UNIX, and Windows、Cloudscape[™] によって提供されるデータベースと、DB2 Connect[™]、DB2 for z/OS、DB2 for i5/OS[®] を介して提供されるデータベースの、メタデータを戻す関数の標準セットを用意しています。

- システム上に PHP 環境をセットアップし、ibm_db2 拡張モジュールを使用可能にする必要があります。
- db2_connect() または db2_pconnect() から戻される接続リソースを持っている必要があります。

1. 必要なメタデータを戻す以下の関数を呼び出します。

db2_client_info()

DB2 クライアント・ソフトウェアおよび構成についてのメタデータを戻します。

db2_column_privileges()

表の列および関連特権をリストします。

db2_columns()

表の列および関連メタデータをリストします。

db2_foreign_keys()

表の外部キーをリストします。

db2_primary_keys()

表の主キーをリストします。

db2_procedure_columns()

1 つ以上のストアド・プロシージャのパラメーターをリストします。

db2_procedures()

データベースに登録されているストアド・プロシージャをリストします。

db2_server_info()

データベース管理システム・ソフトウェアおよび構成についてのメタデータを戻します。

db2_special_columns()

表の固有行 ID をリストします。

db2_statistics()

表の索引および統計をリストします。

db2_table_privileges()

データベース内の表および関連特権をリストします。

ほとんどの `ibm_db2` メタデータ関数は修飾子またはカタログ・パラメーターを受け入れますが、接続するときには、このパラメーターを `NULL` 以外の値のみ設定しなければならないことに注意してください。

2. 呼び出したメタデータ関数に応じて、以下のようになります。

- `db2_client_info()` および `db2_server_info()` 関数は、読み取り専用プロパティを持つ単一のオブジェクトを直接戻します。これらのオブジェクトのプロパティを使用して、接続先のデータベース管理システムに応じて様々な仕方で作動するアプリケーションを作成できます。例えば、`ibm_db2` 拡張モジュールで構築された Web ベースのデータベース管理アプリケーションは、考えられるすべてのデータベース管理システムについて最小公分母の限度をエンコードするのではなく、`db2_server_info()->MAX_COL_NAME_LEN` プロパティを使用して、接続先のデータベース管理システムでの列名の最大長に対応する最大長を持つ命名列のテキスト・フィールドを動的に表示できます。
- その他のメタデータ関数は、各関数について列が定義された結果セットを戻します。この目的のためには、通常の `ibm_db2` 関数を使用して、結果セットから行を取得してください。

メタデータ関数を呼び出すと、かなりの量のデータベース管理システム・リソースを消費することに注意してください。可能であれば、今後の使用のために、呼び出しの結果をキャッシュに入れることを考慮してください。

PHP での XQuery 式の実行 (ibm_db2)

PHP スクリプトは、DB2 データベースに接続した後に、XQuery 式を発行する準備ができます。 `db2_exec()` および `db2_execute()` 関数が SQL ステートメントを実行し、このステートメントにより、XQuery 式を渡すことができます。通常、`db2_exec()` は共通のインクルード・ファイルまたは基本クラスで使用されるアプリケーションのデフォルトのスキーマを設定する際に使用します。

システム上に PHP 環境をセットアップし、`ibm_db2` 拡張モジュールを使用可能にする必要があります。

インジェクション・アタックによるセキュリティ上の脅威を予防するため、`db2_exec()` は静的ストリングで構成される SQL ステートメントを実行する場合にのみ使用する必要があります。ユーザー入力を表す PHP 変数を XQuery 式の中に埋め込むと、アプリケーションがインジェクション・アタックの危険にさらされることがあります。

1. `db2_exec()` に以下の引数を指定して呼び出します。

- a. 接続リソース
- b. SQL ステートメントを含むストリング (XQuery 式を含む)。 SQL ステートメントの `XMLQUERY` 節では XQuery 式を折り返す必要があります。
- c. (オプション): ステートメント・オプションを含む配列

DB2_ATTR_CASE

SQL 標準に準拠しないデータベース・システムとの互換性を保つた

め、このオプションは列名を大/小文字のどちらでアプリケーションに戻すかを設定します。デフォルトでは、`DB2_CASE_NATURAL` に設定され、`DB2` から戻される列名がそのまま戻されます。このパラメーターを、`DB2_CASE_LOWER` に設定して列名を強制的に小文字に変更したり、`DB2_CASE_UPPER` に設定して列名を強制的に大文字に変更することもできます。

DB2_ATTR_CURSOR

このオプションは、`ibm_db2` が結果セットに戻すカーソルの型を設定します。デフォルトで、`ibm_db2` は前方スクロール・カーソル (`DB2_FORWARD_ONLY`) を戻します。この場合、`db2_fetch_array()`、`db2_fetch_assoc()`、`db2_fetch_both()`、`db2_fetch_object()`、または `db2_fetch_row()` の各呼び出しに対して、結果セット内の次の行が戻されます。このパラメーターを `DB2_SCROLLABLE` に設定して両方向スクロール・カーソルを要求することもできます。このとき、`ibm_db2` の `fetch` 関数は、アクセスする結果セット内の行の絶対位置を指定する 2 番目の引数を受け入れるようになります。

2. `db2_exec()` によって戻される値を確認します。
 - 値が `FALSE` の場合、SQL ステートメントは失敗しています。
`db2_stmt_error()` および `db2_stmt_errormsg()` 関数によって診断情報を取得できます。
 - 値が `FALSE` でない場合は、SQL ステートメントが成功してステートメント・リソースが戻されており、この照会に関する後続の関数呼び出しで、戻されたリソースを使用することができます。

```
<?php
$xmlquery = '$doc/customerinfo/phone';
$stmt = db2_exec($conn, "select xmlquery('$xmlquery'
PASSING INFO AS ¥'doc¥') from customer");?>
```

SQL ステートメントの実行

PHP での単一 SQL ステートメントの実行 (ibm_db2)

ほとんどの PHP スクリプトは、`DB2` データベースに接続した後に、1 つ以上の SQL ステートメントを実行します。`db2_exec()` 関数は、入力パラメーターを取らない単一の SQL ステートメントを実行します。通常、`db2_exec()` は共通のインクルード・ファイルまたは基本クラスで使用されるアプリケーションのデフォルトのスキーマを設定する際に使用します。

システム上に PHP 環境をセットアップし、`ibm_db2` 拡張モジュールを使用可能にする必要があります。

SQL インジェクション・アタックによるセキュリティ上の脅威を予防するため、`db2_exec()` は静的ストリングで構成される SQL ステートメントを実行する場合にのみ使用する必要があります。SQL ステートメントへユーザーから入力された PHP 変数を埋め込むことが原因で、アプリケーションが SQL インジェクション・アタックの危険にさらされることがあります。

1. `db2_exec()` に以下の引数を指定して呼び出します。
 - a. 接続リソース
 - b. SQL ステートメントを含むストリング
 - c. (オプション): ステートメント・オプションを含む配列

DB2_ATTR_CASE

SQL 標準に準拠しないデータベース・システムとの互換性を保つため、このオプションは列名を大/小文字のどちらでアプリケーションに戻すかを設定します。デフォルトでは、`DB2_CASE_NATURAL` に設定され、DB2 から戻される列名がそのまま戻されます。このパラメーターを、`DB2_CASE_LOWER` に設定して列名を強制的に小文字に変更したり、`DB2_CASE_UPPER` に設定して列名を強制的に大文字に変更することもできます。

DB2_ATTR_CURSOR

このオプションは、`ibm_db2` が結果セットに戻すカーソルの型を設定します。デフォルトで、`ibm_db2` は前方スクロール・カーソル (`DB2_FORWARD_ONLY`) を戻します。この場合、`db2_fetch_array()`、`db2_fetch_assoc()`、`db2_fetch_both()`、`db2_fetch_object()`、または `db2_fetch_row()` の各呼び出しに対して、結果セット内の次の行が戻されます。このパラメーターを `DB2_SCROLLABLE` に設定して両方向スクロール・カーソルを要求することもできます。このとき、`ibm_db2` の `fetch` 関数は、アクセスする結果セット内の行の絶対位置を指定する 2 番目の引数を受け入れるようになります。

2. `db2_exec()` によって戻される値を確認します。
 - 値が `FALSE` の場合、SQL ステートメントは失敗しています。
`db2_stmt_error()` および `db2_stmt_errormsg()` 関数によって診断情報を取得できます。
 - 値が `FALSE` でない場合は、SQL ステートメントが成功してステートメント・リソースが戻されており、この照会に関する後続の関数呼び出しで、戻されたリソースを使用することができます。

SQL ステートメントが両方向スクロール・カーソルを使用してすでに行を選択しているか、または行を挿入、更新、削除している場合には、`db2_num_rows()` を呼び出してステートメントによって戻された、またはその影響を受けた行数を戻すことができます。SQL ステートメントが結果セットを戻している場合は、行の取り出しを開始できます。

PHP での SQL ステートメントの準備と実行 (ibm_db2)

PHP アプリケーションのほとんどの SQL ステートメントでは、変数入力を使用して SQL ステートメントが完成します。ユーザーによる入力を安全に SQL ステートメントに渡すために、変数の入力を表すパラメーター・マーカー (?) を使用したステートメントを準備します。そのようにして作成された準備済みステートメントは、実行時に入力値をパラメーター・マーカーにバインドすることで SQL が完成します。バインドされた値は、データベース・エンジンにより単一のパラメータとして扱われ、これによってアプリケーションに対する SQL インジェクション・ア

タックを予防します。同じステートメントを繰り返し実行する場合、`db2_exec()` によって発行されるステートメントと比較すると、準備済みステートメントの方がパフォーマンスの面で優れています。なぜなら、準備済みステートメントはデータベース管理システムが作成時に一度だけ用意したアクセス・プランを再利用できるためです。

システム上に PHP 環境をセットアップし、`ibm_db2` 拡張モジュールを使用可能にする必要があります。

パラメーター・マーカーは、列の値または述部の値のプレースホルダーとしてのみ使用できます。列名、表名、またはその他の SQL ID の代わりにパラメーター・マーカーが使用されているステートメントについては、SQL コンパイラーでアクセス・プランを作成することはできません。

SQL ステートメントを準備および実行するには、次のようにします。

1. `db2_prepare()` に以下の引数を指定して呼び出します。
 - a. 接続リソース
 - b. SQL ステートメントを含むストリング (変数入力を必要とする列値または述部値のパラメーター・マーカー (?) を含む)
 - c. (オプション): ステートメント・オプションを含む配列

DB2_ATTR_CASE

SQL 標準に準拠しないデータベース・システムとの互換性を保つため、このオプションは列名を大/小文字のどちらでアプリケーションに戻すかを設定します。デフォルトでは、`DB2_CASE_NATURAL` に設定され、`DB2` から戻される列名がそのまま戻されます。このパラメーターを、`DB2_CASE_LOWER` に設定して列名を強制的に小文字に変更したり、`DB2_CASE_UPPER` に設定して列名を強制的に大文字に変更することもできます。

DB2_ATTR_CURSOR

このオプションは、`ibm_db2` が結果セットに戻すカーソルの型を設定します。デフォルトで、`ibm_db2` は前方スクロール・カーソル (`DB2_FORWARD_ONLY`) を戻します。この場合、`db2_fetch_array()`、`db2_fetch_assoc()`、`db2_fetch_both()`、`db2_fetch_object()`、または `db2_fetch_row()` の各呼び出しに対して、結果セット内の次の行が戻されます。このパラメーターを `DB2_SCROLLABLE` に設定して両方向スクロール・カーソルを要求することもできます。このとき、`ibm_db2` の `fetch` 関数は、アクセスする結果セット内の行の絶対位置を指定する 2 番目の引数を受け入れるようになります。

2. `db2_prepare()` によって戻される値を確認します。
 - 値が `FALSE` の場合、SQL ステートメントは失敗しています。`db2_stmt_error()` および `db2_stmt_errormsg()` 関数によって診断情報を取得できます。
 - 値が `FALSE` でない場合は、SQL ステートメントが成功してステートメント・リソースが戻されており、この照会に関する後続の関数呼び出しで、戻されたリソースを使用することができます。

3. (オプション): SQL ステートメント内の各パラメーター・マーカーに対して、`db2_bind_param()` に次の引数を指定して呼び出します。
 - a. ステートメント・リソース
 - b. SQL ステートメント内でのパラメーター・マーカーの位置を表す整数
 - c. パラメーター・マーカーの代わりに使用する値
4. `db2_execute` に以下の引数を指定して呼び出します。
 - a. ステートメント・リソース
 - b. (オプション): パラメーター・マーカーの代わりに使用する値を順番に含む配列

```
$sql = "SELECT firstme, lastname FROM employee WHERE bonus > ? AND bonus < ?";
$stmt = db2_prepare($conn, $sql);
if (!$stmt) {
    // Handle errors
}
```

```
// Explicitly bind parameters
db2_bind_param($stmt, 1, $_POST['lower']);
db2_bind_param($stmt, 2, $_POST['upper']);
```

```
db2_execute($stmt);
// Process results
```

```
// Invoke prepared statement again using dynamically bound parameters
db2_execute($stmt, array($_POST['lower'], $_POST['upper']));
```

1 つ以上の結果セットを戻す準備済みステートメントを実行する場合、`db2_fetch_array()`、`db2_fetch_assoc()`、`db2_fetch_both()`、`db2_fetch_object()`、または `db2_fetch_row()` 関数を呼び出すことにより、ステートメント・リソースからの行の取得を開始できます。

PHP でのラージ・オブジェクトの挿入 (ibm_db2)

`ibm_db2` 拡張モジュールはすべての DB2 データ・タイプをサポートします。これには、文字ラージ・オブジェクト (CLOB) およびバイナリー・ラージ・オブジェクト (BLOB) データ・タイプが含まれます。ラージ・オブジェクトをデータベースに挿入すると、そのラージ・オブジェクトを単に PHP ストリングとして扱うことができます。しかし、ラージ・オブジェクトを PHP ストリングとして扱う方法では、PHP サーバー上で必要以上にリソースを消費することになります。ラージ・オブジェクトのすべてのデータを PHP ストリングにロードしてから `INSERT` ステートメントを使用してそれを DB2 に渡す代わりに、PHP サーバー上のファイルから直接ラージ・オブジェクトを挿入できます。

システム上に PHP 環境をセットアップし、`ibm_db2` 拡張モジュールを使用可能にする必要があります。

ラージ・オブジェクトをファイルから直接データベースに挿入するには、次のようになしてください。

1. `db2_prepare()` を呼び出して、ラージ・オブジェクト列を表すパラメーター・マーカーを持つ `INSERT` ステートメントを準備します。

2. PHP 変数の値を、ラージ・オブジェクトのデータを含むファイルのパスおよび名前に設定します。パスは相対でも絶対でも構いません。また、PHP 実行可能プログラムからアクセス可能であることが必要です。
3. `db2_bind_param()` を呼び出して、ラージ・オブジェクトのデータを含むファイルに、パラメーター・マーカをバインドします。3 番目のパラメーターは、ラージ・オブジェクトのデータを含むファイルの名前を保持する PHP 変数の名前を表す文字列です。4 番目のパラメーターは `DB2_PARAM_FILE` です。これは、データをファイルから取得するように `ibm_db2` 拡張モジュールに通知します。
4. `db2_execute()` を呼び出して、`INSERT` ステートメントを発行し、ファイルからのデータをデータベースにバインドします。

```
$stmt = db2_prepare($conn, "INSERT INTO animal_pictures(picture) VALUES (?");
$picture = "/opt/albums/spook/grooming.jpg";
$rc = db2_bind_param($stmt, 1, "picture", DB2_PARAM_FILE);
$rc = db2_execute($stmt);
```

照会結果セットの読み取り

PHP での結果セットからの列のフェッチ (`ibm_db2`)

1 つ以上の結果セットを戻すステートメントを実行する場合、通常はそれぞれの結果セット上で戻された行を繰り返して取り出すことが必要になります。結果セットに非常に大きいデータを含む列 (`BLOB` または `CLOB` データ・タイプで定義された列など) が含まれる場合、列ごとにデータを取得して、PHP 処理で過剰なメモリーを使用するのを避けることができます。

- システム上に PHP 環境をセットアップし、`ibm_db2` 拡張モジュールを使用可能にする必要があります。
 - `db2_exec()` または `db2_execute()` によって戻され、関連する 1 つ以上の結果セットを持つステートメント・リソースが必要です。
1. `db2_fetch_row()` 関数を呼び出して、結果セット内の次の行にカーソルを進めます。特定の結果セットについて初めて `fetch` 関数を呼び出すときに、カーソルは結果セットの最初の行に進みます。両方向スクロール・カーソルを要求した場合、結果セット内で取得する行の数も指定できます。
 2. `db2_fetch_row()` によって戻される結果を確認します。結果が `FALSE` である場合、結果セット内に行がなくなったことを示します。
 3. `db2_result()` 関数を呼び出して、行の中の列の位置を表す整数 (最初の列は 0 で始まる) か、または列の名前を表す文字列を渡すことによって、要求された列からの値を取得します。

```
<?php
$sql = 'SELECT name, breed FROM animals WHERE weight < ?';
$stmt = db2_prepare($conn, $sql);
db2_execute($stmt, array(10));
while (db2_fetch_row($stmt)) {
    $name = db2_result($stmt, 0);
    $breed = db2_result($stmt, 'BREED');
    print "$name $breed";
}
?>
```

PHP での結果セットからの行の取り出し (ibm_db2)

1 つ以上の結果セットを戻すステートメントを実行した場合、通常、返された複数の行を順に取り出す作業が必要になります。

- システム上に PHP 環境をセットアップし、ibm_db2 拡張モジュールを使用可能にする必要があります。
- db2_exec() または db2_execute() によって戻され、関連する 1 つ以上の結果セットを持つステートメント・リソースが必要です。

希望する形式で行のデータを戻す ibm_db2 の fetch 関数を呼び出します。

db2_fetch_array()

0 から始まる値で番号付けされた列に対応するデータを含む配列を返します。

db2_fetch_assoc()

列名を指定すると、その列のデータを含む配列を返します。

db2_fetch_both()

列名、もしくは 0 から始まる番号で表された各列に対応するデータを含む配列を返します。

db2_fetch_object()

行からのデータを含むオブジェクトを返します。オブジェクトには行の列名と一致するプロパティが保持され、アクセスすると各列の対応する値が返されます。

ステートメント・リソースは最初の引数として渡す必要があります。db2_exec() または db2_prepare() の実行時に両方向スクロール・カーソルを要求した場合、絶対行番号を 2 番目の引数として渡すことができます。デフォルトの前方スクロール・カーソルを使用すると、取り出しメソッドを呼び出すたびに、結果セット内の次の行が戻されます。結果セットの終わりに達したことを示す FALSE を取り出しメソッドが戻すまで、行の取り出しを継続できます。

```
$stmt = db2_exec($conn, "SELECT firstnme, lastname FROM employee");
while ($row = db2_fetch_object($stmt)) {
    print "Name: <p>{$row->FIRSTNME} {$row->LASTNAME}</p>";
}
```

PHP でのラージ・オブジェクトのフェッチ (ibm_db2)

ibm_db2 拡張モジュールはすべての DB2 データ・タイプをサポートします。これには、文字ラージ・オブジェクト (CLOB) およびバイナリー・ラージ・オブジェクト (BLOB) データ・タイプが含まれます。ラージ・オブジェクトを結果セットから取り出すと、そのラージ・オブジェクトを単に PHP ストリングとして扱うことができます。しかし、ラージ・オブジェクトを PHP ストリングとして扱う方法では、PHP サーバー上で必要以上にリソースを消費することになります。ラージ・オブジェクトのデータを含むファイルを作成することが最終的な目標である場合、ラージ・オブジェクトを PHP サーバー上のファイルに直接取り出すことにより、システム・リソースを節約することができます。

システム上に PHP 環境をセットアップし、ibm_db2 拡張モジュールを使用可能にする必要があります。

ラージ・オブジェクトをデータベースからファイルに直接取り出すには、次のようにしてください。

1. ストリームを表す PHP 変数を作成します。例えば、`fopen()` への呼び出しからの戻り値です。
2. `db2_prepare()` を呼び出して、`SELECT` ステートメントを作成します。
3. `db2_bind_param()` を呼び出して、ラージ・オブジェクトの出力列を、ストリームを表す PHP 変数にバインドします。3 番目のパラメーターは、ラージ・オブジェクトのデータを含むファイルの名前を保持する PHP 変数の名前を表す文字列です。4 番目のパラメーターは `DB2_PARAM_FILE` です。これは、データをファイルに書き込むように `ibm_db2` 拡張に通知します。
4. `db2_execute()` を呼び出して、`SQL` ステートメントを発行します。
5. 選択した `ibm_db2` 取り出し関数（例えば、`db2_fetch_object()`）を呼び出して、結果セット内の次の行を検索します。

```
$stmt = db2_prepare($conn, "SELECT name, picture FROM animal_pictures");  
$picture = fopen("/opt/albums/spook/grooming.jpg", "wb");  
$rc = db2_bind_param($stmt, 1, "nickname", DB2_CHAR, 32);  
$rc = db2_bind_param($stmt, 2, "picture", DB2_PARAM_FILE);  
$rc = db2_execute($stmt);  
$rc = db2_fetch_object($stmt);
```

PHP でのトランザクションの管理 (ibm_db2)

`ibm_db2` 拡張モジュールは、デフォルトで、すべての接続を自動コミット・モードで開きます。自動コミット・モードは、スケラビリティが必要な Web アプリケーションのパフォーマンスを阻害するロック・エスカレーションの問題を防ぐのに役立ちます。しかし、スクリプトによっては、1 つ以上の `SQL` ステートメントを含むトランザクションのロールバックが必要になる場合があります。`ibm_db2` 拡張モジュールを使用することによって、トランザクションをきめ細かく制御することができます。

システム上に PHP 環境をセットアップし、`ibm_db2` 拡張モジュールを使用可能にする必要があります。

PHP でデータベース・トランザクションを制御するには、必ず `db2_connect()` で作成した通常の接続を使用します。持続的な接続では、常に自動コミット・モードが使用されるからです。

トランザクションを開始するには、次のようにします。

1. `db2_connect()` オプション配列に `"AUTOCOMMIT" => DB2_AUTOCOMMIT_OFF` 設定を使用して、データベース接続を作成します。`db2_autocommit($conn, DB2_AUTOCOMMIT_OFF)` を呼び出して、既存の接続リソースに対して自動コミットをオフにすることもできます。`db2_autocommit()` を呼び出すと、PHP からデータベース管理システムへの通信が追加で必要になり、PHP スクリプトのパフォーマンスに影響する場合があります。
2. トランザクションが使用可能にされている接続リソースを使用して、データベース・トランザクションの有効範囲内で 1 つ以上の `SQL` ステートメントを発行します。
3. トランザクションをコミットまたはロールバックします。

- トランザクションをコミットするには、`db2_commit()` を呼び出します。
 - トランザクションをロールバックするには、`db2_rollback()` を呼び出します。
4. (オプション): `db2_autocommit($conn, DB2_AUTOCOMMIT_ON)` を呼び出して、データベース接続を自動コミット・モードに戻します。データベース接続を自動コミット・モードに戻さずに別の SQL ステートメントを発行すると、新規トランザクションが開始し、コミットやロールバックの処理が必要になります。

SQL ステートメントをトランザクション内で発行し、明示的にトランザクションをコミットまたはロールバックせずにスクリプトが終了する場合、`ibm_db2` 拡張モジュールはトランザクション内で実行されたすべての作業を自動的にロールバックします。

```
$conn = db2_connect('SAMPLE', 'db2inst1', 'ibmdb2', array(
    'AUTOCOMMIT' => DB2_AUTOCOMMIT_ON));

// Issue one or more SQL statements within the transaction
$result = db2_exec($conn, 'DELETE FROM TABLE employee');
if ($result === FALSE) {
    print '<p>Unable to complete transaction!</p>';
    db2_rollback($conn);
}
else {
    print '<p>Successfully completed transaction!</p>';
    db2_commit($conn);
}
```

エラーおよび警告メッセージの処理 (ibm_db2)

データベースに接続しようとしたり、SQL ステートメントを発行しようとする、問題が発生することがあります。接続用のパスワードが正しくないか、`SELECT` ステートメントで参照した表が存在しないか、または SQL ステートメントの構文が無効である可能性があります。コード化は慎重に行い、`ibm_db2` 拡張モジュールで提供されるエラー処理関数を使用して、アプリケーションに問題があった時に、正常にリカバリーできるようにする必要があります。

システム上に PHP 環境をセットアップし、`ibm_db2` 拡張モジュールを使用可能にする必要があります。

1. `ibm_db2` 関数から戻される値を調べて、関数が正常に戻されたことを確認します。関数が値 0 を戻すことができる場合 (`db2_num_rows()` など)、PHP の `===` 演算子を使用して、値が `FALSE` だったかどうかを明示的にテストする必要があります。
2. 関数が、予期していた接続リソース、ステートメント・リソース、または数値の代わりに `FALSE` を戻した場合、アプリケーションのコンテキストおよびアプリケーションの必要に適した `ibm_db2` エラー処理関数を呼び出します。

接続エラー

最後の接続試行で戻された `SQLSTATE` を検索するには、`db2_conn_error()` を呼び出します。アプリケーション・エラー・ログに該当する記述エラー・メッセージを検索するには、`db2_conn_errormsg()` を呼び出します。

```

$connection = db2_connect($database, $user, $password);
if (!$connection) {
    $this->state = db2_conn_error();
    return false;
}

```

SQL エラー (SQL ステートメントを直接実行し、結果を取り出す)

SQL ステートメントを準備または実行する最後の試行で戻された SQLSTATE を検索するか、または結果セットから結果を取り出すには、db2_stmt_error() を呼び出します。アプリケーション・エラー・ログに該当する記述エラー・メッセージを検索するには、db2_stmt_errormsg() を呼び出します。

```

$stmt = db2_prepare($connection, "DELETE FROM employee
WHERE firstnme = ?");
if (!$stmt) {
    $this->state = db2_stmt_error();
    return false;
}

```

SQL エラー (準備済みステートメントを実行)

db2_prepare() が正常に戻されたが、db2_execute() に対する後続の呼び出しが失敗する場合、db2_stmt_error() または db2_stmt_errormsg() を呼び出し、db2_prepare() への呼び出しから戻されたリソースを引数として渡します。

```

$success = db2_execute($stmt, array('Dan'));
if (!$success) {
    $this->state = db2_stmt_error($stmt);
    return $false;
}

```

3. データベースから戻された生の SQLSTATE を直接表示することに起因するセキュリティの脆弱性が生じる可能性を防ぎ、Web アプリケーションでの全体的なユーザー・エクスペリエンスを向上させるには、switch 構造を使用して、既知のエラー状態から回復するか、カスタム・エラー・メッセージを戻します。

```

switch($this->state):
case '22001':
    // More data than allowed for the defined column
    $message = "You entered too many characters for this value.";
    break;

```

ストアド・プロシージャの呼び出し

PHP での、OUT または INOUT パラメーターを持つストアド・プロシージャの呼び出し (ibm_db2)

DB2 は、入力値を受け入れるだけのパラメーター (IN パラメーター)、出力値を戻すだけのパラメーター (OUT パラメーター)、または入力値を受け入れて出力値を戻すパラメーター (INOUT) を持つストアド・プロシージャをサポートします。PHP 用の ibm_db2 拡張モジュールを使用すると、SQL ステートメント内のその他のパラメーター・マーカースと同様に IN パラメーターを処理できます。しかし、ibm_db2 拡張モジュールでは、OUT および INOUT パラメーターを持つストアド・プロシージャを呼び出し (CALL)、それらのパラメーターから出力値を取得することもできます。

システム上に PHP 環境をセットアップし、ibm_db2 拡張モジュールを使用可能にする必要があります。

OUT または INOUT パラメーターを持つストアード・プロシージャを呼び出すには、次のようにしてください。

1. db2_prepare() を呼び出して、OUT および INOUT パラメーターを表すパラメーター・マーカを持つ CALL ステートメントを準備します。
2. db2_bind_param() を呼び出して、CALL ステートメントが発行された後にパラメーターの出力値を保持する PHP 変数の名前にそれぞれのパラメーター・マーカをバインドします。INOUT パラメーターの場合、PHP 変数の値は、CALL ステートメントが発行されるたびにパラメーターの入力値として渡されます。db2_bind_param() の 4 番目のパラメーターを、DB2_PARAM_OUT (OUT パラメーターを表す) または DB2_PARAM_INOUT (INOUT パラメーターを表す) のいずれかに設定します。
3. db2_execute() を呼び出して、CALL ステートメントを発行し、ストアード・プロシージャからのデータを PHP 変数にバインドします。

```
$sql = 'CALL match_animal(?, ?)';
$stmt = db2_prepare($conn, $sql);

$second_name = "Rickety Ride";
$weight = 0;

db2_bind_param($stmt, 1, "second_name", DB2_PARAM_INOUT);
db2_bind_param($stmt, 2, "weight", DB2_PARAM_OUT);

print "Values of bound parameters _before_ CALL:¥n";
print " 1: {$second_name} 2: {$weight}¥n";

db2_execute($stmt);

print "Values of bound parameters _after_ CALL:¥n";
print " 1: {$second_name} 2: {$weight}¥n";
```

PHP での、複数の結果セットを戻すストアード・プロシージャの呼び出し (ibm_db2)

DB2 では、複数の結果セットを戻すストアード・プロシージャの作成および呼び出しを行うことができます。PHP 用の ibm_db2 拡張モジュールは、db2_next_result() 関数によって、この機能を完全にサポートします。この関数を使用して、同じストアード・プロシージャに対する単一の呼び出しで戻されるさまざまな結果セットから、希望する順序で行を取り出すことができます。

- システム上に PHP 環境をセットアップし、ibm_db2 拡張モジュールを使用可能にする必要があります。
- db2_exec() または db2_execute() を使用してストアード・プロシージャを呼び出すことによって戻されるステートメント・リソースが必要です。

ストアード・プロシージャから複数の結果セットを戻すには、次のようにしてください。

1. 最初の結果セットは、CALL ステートメントによって戻されるステートメント・リソースに関連付けられます。

- オリジナルのステートメント・リソースを最初の引数として `db2_next_result()` に渡し、2 番目以降の結果セットを取得します。取得できる結果セットがなくなると、この関数は `FALSE` を返します。

```
$stmt = db2_exec($conn, 'CALL multiResults()');

print "Fetching first result set\n";
while ($row = db2_fetch_array($stmt)) {
    // work with row
}

print "\nFetching second result set\n";
$result_2 = db2_next_result($stmt);
if ($result_2) {
    while ($row = db2_fetch_array($result_2)) {
        // work with row
    }
}

print "\nFetching third result set\n";
$result_3 = db2_next_result($stmt);
if ($result_3) {
    while ($row = db2_fetch_array($result_3)) {
        // work with row
    }
}
```

第 4 章 PDO を使用した開発

PHP による DB2 データベースへの接続 (PDO)

DB2 データ・ソースに対してデータの作成、更新、削除、または検索を行うには、あらかじめ DB2 データベースに接続する必要があります。PHP 用の PHP Data Objects (PDO) インターフェースにより、PDO_IBM 拡張モジュールを介し、DB2 データベース管理システムへのカタログ式接続または直接 TCP/IP 接続を使用して DB2 データベースに接続することができます。また、データ・ソースへの持続的な接続を行うこともできます。PHP からデータベースへの接続をオープンしたままにしておき、同一セットの ID とパスワードを持つ PHP スクリプトが接続を要求したときにその接続を再利用するという方法によって、パフォーマンスを向上させます。

システム上に PHP 5.1 以上の環境をセットアップし、PDO および PDO_IBM 拡張モジュールを使用可能にする必要があります。

1. try{} ブロック内で PDO コンストラクターを呼び出して、DB2 データベースへの接続を作成します。DSN 値は PDO_IBM 拡張モジュール用のものを使用します。まず `ibm:` から始め、その次にカタログしたデータベース名または TCP/IP 直接接続に使用する完全なデータベース接続ストリングのどちらかを指定します。
 - (Windows): デフォルトで、PDO_IBM は接続プールを使用し、接続リソースをできるだけ少なくして、接続パフォーマンスを向上させます。
 - (Linux および UNIX): `array(PDO::ATTR_PERSISTENT => TRUE)` を 4 番目の引数として PDO コンストラクターに渡した場合、PDO_IBM は持続的な接続を提供します。
2. (オプション): PDO コンストラクターに渡す 4 番目の引数に、PDO 接続のエラー処理オプションを以下のように設定します。
 - PDO は、エラー発生時に `PDO::errorInfo()` で取得されるエラー・メッセージと `PDO::errorCode()` で取得される SQLCODE をデフォルトで設定します。このモードを明示的に要求するには、`PDO::ATTR_ERRMODE => PDO::ERRMODE_SILENT` を設定します。
 - エラーが発生したときに PHP の `E_WARNING` が出されるようにするには、エラー・メッセージと SQLCODE の設定に加えて、`PDO::ATTR_ERRMODE => PDO::ERRMODE_WARNING` を設定します。
 - エラー発生時に PHP 例外がスローされるようにするには、`PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION` を設定します。
3. try{} ブロックでスローされた例外を、対応する catch {} ブロックでキャッチします。

```
try {
    $connection = new PDO("ibm:SAMPLE", "db2inst1", "ibmdb2", array(
        PDO::ATTR_PERSISTENT => TRUE,
        PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION)
    );
```

```

    }
    catch (Exception $e) {
        echo($e->getMessage());
    }
}

```

PDO を介して接続を作成した場合、以下の操作を行ったときに、PHP はデータベースへの接続を閉じます。

- PDO オブジェクトを NULL に設定する
- または、PHP スクリプトが終了する

SQL ステートメントの実行

PHP での、結果セットを戻さない単一 SQL ステートメントの実行 (PDO)

ほとんどの PHP スクリプトは、DB2 データベースに接続した後に、1 つ以上の SQL ステートメントを実行します。PDO::exec() メソッドは、入力パラメーターを取らず結果セットを戻さない単一の SQL ステートメントを実行します。通常、PDO::exec() は共通のインクルード・ファイルまたは基本クラスで使用されるアプリケーションのデフォルトのスキーマを設定する際に使用します。

システム上に PHP 環境をセットアップし、PDO_IBM 拡張モジュールを使用可能にする必要があります。

SQL インジェクション・アタックによるセキュリティ上の脅威を予防するため、PDO::exec() は静的ストリングで構成される SQL ステートメントを実行する場合にのみ使用する必要があります。SQL ステートメントへユーザーから入力された PHP 変数を埋め込むことが原因で、アプリケーションが SQL インジェクション・アタックの危険にさらされることがあります。

PHP で作成された単一の SQL ステートメントを実行するには、次のようにします。

1. PDO 接続オブジェクト上で、PDO::exec() メソッドに SQL ステートメントを含むストリングを指定して呼び出します。
2. SQL ステートメントによって行が挿入、変更、または削除されている場合、PDO::exec() は挿入、変更、または削除された行数を表す整数値を返します。PDO::exec() がエラー状態を表す FALSE または行が挿入、変更、削除されていないことを示す 0 を戻したかどうかを判別するには、=== 演算子を使用して FALSE に対して戻り値を厳密にテストする必要があります。

```

$conn = new PDO('ibm:SAMPLE', 'db2inst1', 'ibmdb2');
$result = $conn->exec('SET SCHEMA myapp');
if ($result === FALSE) {
    print "Failed to set schema: " . $conn->errorMsg();
}

```

PHP での、結果セットを戻す単一 SQL ステートメントの実行 (PDO)

ほとんどの PHP スクリプトは、DB2 データベースに接続した後に、1 つ以上の SQL ステートメントを実行します。PDO::query() メソッドは、入力パラメーターを取らず 1 つ以上の結果セットを戻す単一の SQL ステートメントを実行します。通常、PDO::query() は、静的 SELECT ステートメントを実行するときに使用します。

システム上に PHP 環境をセットアップし、PDO_IBM 拡張モジュールを使用可能にする必要があります。

SQL インジェクション・アタックによるセキュリティ上の脅威を予防するため、PDO::query() は静的ストリングで構成される SQL ステートメントを実行する場合にのみ使用する必要があります。SQL ステートメントへユーザーから入力された PHP 変数を埋め込むことが原因で、アプリケーションが SQL インジェクション・アタックの危険にさらされることがあります。

PHP で記述された、結果セットを戻す単一 SQL ステートメントを実行するには、次のようにします。

1. PDO 接続オブジェクト上で、PDO::query() メソッドに SQL ステートメントを含むストリングを指定して呼び出します。
2. PDO::query() によって戻された値を確認します。
 - 値が FALSE の場合、SQL ステートメントは失敗しています。PDO::errorCode() および PDO::errorInfo() メソッドによって診断情報を取得できます。
 - 値が FALSE ではない場合は、SQL ステートメントが成功し、後続のメソッド呼び出しで使用できる PDOStatement リソースが戻されています。

```
$conn = new PDO('ibm:SAMPLE', 'db2inst1', 'ibmdb2');
$result = $conn->query('SELECT firstnme, lastname FROM employee');
if (!$result) {
    print "<p>Could not retrieve employee list: " . $conn->errorMsg(). "</p>";
}
while ($row = $conn->fetch()) {
    print "<p>Name: {$row[0] $row[1]}</p>";
}
```

PDO::query() で PDOStatement オブジェクトを得た後は、PDOStatement::fetch() または PDOStatement::fetchAll() メソッドを使用してオブジェクトから行の取得が可能です。

SQL ステートメントの準備と実行 (PDO)

PHP アプリケーションのほとんどの SQL ステートメントでは、変数入力を使用して SQL ステートメントが完成します。ユーザーによる入力を安全に SQL ステートメントに渡すために、変数の入力を表すパラメーター・マーカー (?) または名前付き変数を使用したステートメントを準備します。そのようにして作成された準備済みステートメントは、実行時に入力値をパラメーター・マーカーにバインドすることで SQL が完成します。バインドされた値は、データベース・エンジンにより単一のパラメータとして扱われ、これによってアプリケーションに対する SQL イン

ジェクション・アタックを予防します。同じステートメントを繰り返し実行する場合、PDO::exec() によって発行されるステートメントと比較すると、準備済みステートメントの方がパフォーマンスの面で優れています。なぜなら、準備済みステートメントはデータベース管理システムが作成時に一度だけ用意したアクセス・プランを再利用できるためです。

システム上に PHP 環境をセットアップし、PDO_IBM 拡張モジュールを使用可能にする必要があります。

- パラメーター・マーカーは、列の値または述部の値のプレースホルダーとしてのみ使用できます。列名、表名、またはその他の SQL ID の代わりにパラメーター・マーカーが使用されているステートメントについては、SQL コンパイラーでアクセス・プランを作成することはできません。
- 疑問符パラメーター・マーカー (?) と名前付きパラメーター・マーカー (:name) を同じ SQL ステートメント内で同時に使用することはできません。

SQL ステートメントを準備および実行するには、次のようにします。

1. PDO::prepare() に以下の引数を指定して呼び出します。
 - a. SQL ステートメントを含むストリング (変数入力が必要とする列値または述部値のパラメーター・マーカー (?) または名前付き変数 (:name) を含む)
 - b. (オプション): ステートメント・オプションを含む配列

PDO::ATTR_CURSOR

このオプションは、PDO が結果セットに戻すカーソルの型を設定します。デフォルトで、PDO は前方スクロール・カーソル (PDO::CURSOR_FWDONLY) を戻します。この場合、PDOStatement::fetch() の各呼び出しに対して、結果セット内の次の行が戻されます。このパラメーターを PDO::CURSOR_SCROLL に設定して、両方向スクロール・カーソルを要求することもできます。

2. PDO::prepare() によって戻された値を確認します。
 - 値が FALSE の場合、SQL ステートメントは失敗しています。PDO::errorCode() および PDO::errorInfo() メソッドによって診断情報を取得できます。
 - 値が FALSE ではない場合は、SQL ステートメントが成功し、後続のメソッド呼び出しで使用可能な PDOStatement オブジェクトが戻されています。
3. (オプション): SQL ステートメント内の各パラメーター・マーカーに対して、PDOStatement::bindParam() に次の引数を指定して呼び出します。
 - a. パラメーター ID。疑問符パラメーター・マーカー (?) の場合、これは SQL ステートメント内のパラメーターの 1 から順に番号付けされた位置を表す整数です。名前付きパラメーター・マーカー (:name) の場合、これはパラメーター名を表すストリングです。
 - b. パラメーター・マーカーの代わりに使用する値
4. PDOStatement::execute() を呼び出します。パラメーター・マーカーの代わりに、使用する値を含む配列を渡すことも可能で、疑問符パラメーター・マーカーの代わりに使用する場合は値を順番に指定し、名前付きパラメーター・マーカーの代わりに使用する場合は :name => value 連想配列として配列を渡します。

```

$sql = "SELECT firstnme, lastname FROM employee WHERE bonus > ? AND bonus < ?";
$stmt = $conn->prepare($sql);
if (!$stmt) {
    // Handle errors
}

// Explicitly bind parameters
$stmt->bindParam(1, $_POST['lower']);
$stmt->bindParam(2, $_POST['upper']);

$stmt->execute($stmt);

// Invoke statement again using dynamically bound parameters
$stmt->execute($stmt, array($_POST['lower'], $_POST['upper']));

```

1 つ以上の結果セットを戻す準備済みステートメントの実行に成功した場合、`PDOStatement::fetch()` または `PDOStatement::fetchAll()` メソッドを呼び出すことにより、ステートメント・リソースからの行の取得を開始できます。

PHP でのラージ・オブジェクトの挿入 (PDO)

PDO 拡張モジュールはすべての DB2 データ・タイプをサポートします。これには、文字ラージ・オブジェクト (CLOB) およびバイナリー・ラージ・オブジェクト (BLOB) データ・タイプが含まれます。ラージ・オブジェクトをデータベースに挿入すると、そのラージ・オブジェクトを単に PHP ストリングとして扱うことができます。しかし、ラージ・オブジェクトを PHP ストリングとして扱う方法では、PHP サーバー上で必要以上にリソースを消費することになります。ラージ・オブジェクトのすべてのデータを PHP ストリングにロードしてから INSERT ステートメントを使用してそれを DB2 に渡す代わりに、PHP サーバー上のファイルから直接ラージ・オブジェクトを挿入できます。

システム上に PHP 5.1 以上の環境をセットアップし、PDO および PDO_IBM 拡張モジュールを使用可能にする必要があります。

ラージ・オブジェクトをファイルから直接データベースに挿入するには、次のようにしてください。

1. `PDO::prepare()` を呼び出して、ラージ・オブジェクト列を表すパラメーター・マーカータを持つ INSERT ステートメントから `PDOStatement` オブジェクトを作成します。
2. ストリームを表す PHP 変数 (例えば、`fopen()` への呼び出しからの戻り値) を作成します。
3. `PDOStatement::bindParam()` を呼び出して、ラージ・オブジェクトのデータのストリームを表す PHP 変数に、パラメーター・マーカータをバインドします。3 番目のパラメーターは、ラージ・オブジェクトのデータを含むファイルの名前を保持する PHP 変数の名前を表すストリングです。4 番目のパラメーターは PHP 定数、`PDO::PARAM_LOB` です。これは、データをファイルから取り出すように PDO 拡張モジュールに通知します。
4. `PDOStatement::execute()` を呼び出して、INSERT ステートメントを発行し、ファイルからのデータをデータベースにバインドします。

```

$stmt = $conn->prepare("INSERT INTO animal_pictures(picture) VALUES (?)");
$picture = fopen("/opt/albums/spook/grooming.jpg", "rb");
$stmt->bindParam($stmt, 1, $picture, PDO::PARAM_LOB);
$stmt->execute();

```

照会結果セットの読み取り

PHP での結果セットからの列のフェッチ (PDO)

1 つ以上の結果セットを戻すステートメントを実行する場合、通常はそれぞれの結果セット上で戻された行を繰り返して取り出すことが必要になります。結果セット内の各行から単一の列だけを取り出したいという場合があります。必要な列のみを返すように SELECT ステートメントを書き換えることができる場合がありますが、必要な数より多くの列を戻すストアード・プロシージャについては、それを書き換えるための特権を持っていない場合もあるでしょう。

- システム上に PHP 環境をセットアップし、PDO および PDO_IBM 拡張モジュールを使用可能にする必要があります。
- `PDO::query()` または `PDOStatement::execute()` によって戻され、1 つ以上の関連した結果セットを持つステートメント・リソースが必要です。

行全体からすべての列を同時に検索するのではなく、行から 1 列ずつ取り出すことに決めた場合、各行から単一列だけを戻すことができます。

- 結果セット内の単一行から単一列を戻すには、次のようにしてください。

取り出す列をメソッドの最初の引数で指定して、`PDOStatement::fetchColumn()` メソッドを呼び出します。列の番号は 0 で始まります。列を指定しない場合、`PDOStatement::fetchColumn()` は行の中の最初の列を戻します。

- 結果セット内にある残りの行のすべてから単一の列を含む配列を戻すには、次のようにしてください。

`PDO::FETCH_COLUMN` を最初の引数として渡し、検索する列を 2 番目の引数として渡して、`PDOStatement::fetchAll()` メソッドを呼び出し、結果セットから選択した列の値の配列を戻します。列の番号は 0 で始まります。列を指定しない場合、`PDOStatement::fetchAll(PDO::FETCH_COLUMN)` は行の中の最初の列を戻します。

PHP での結果セットからの行の取り出し (PDO)

1 つ以上の結果セットを戻すステートメントを実行した場合、通常、返された複数の行を順に取り出す作業が必要になります。

- システム上に PHP 環境をセットアップし、PDO 拡張モジュールを使用可能にする必要があります。
- `PDO::query()` または `PDOStatement::execute()` によって戻され、1 つ以上の関連した結果セットを持つ `PDOStatement` オブジェクトが必要です。

結果セットから単一行を配列またはオブジェクトとして戻すには、`PDOStatement::fetch()` メソッドを呼び出します。

結果セットからすべての行を配列の配列またはオブジェクトの配列として戻すには、`PDOStatement::fetchAll()` メソッドを呼び出します。

デフォルトで、PDO は各行を、列名と行内の列位置によって 0 から順に番号付けされた配列として戻します。PDOStatement::fetch() の最初のパラメーターとして次の定数のいずれかを渡すことにより、配列が異なったスタイルで戻されるよう指定することができます。

PDO::FETCH_ASSOC

結果セット内に戻される際の列名によってインデックス付けされた配列を戻します。

PDO::FETCH_BOTH (デフォルト)

結果セット内に戻される際の列名と 0 から順に番号付けされた列番号の両方によってインデックス付けされた配列を戻します。

PDO::FETCH_BOUND

TRUE を戻し、結果セット内の列の値を、PDOStatement::bindParam() メソッドによってバインドされた PHP 変数に設定します。

PDO::FETCH_CLASS

要求されたクラスの新規インスタンスを戻し、結果セットの列をそのクラスの名前付きプロパティにマップします。

PDO::FETCH_INTO

要求されたクラスの既存のインスタンスを更新し、結果セットの列をそのクラスの名前付きプロパティにマップします。

PDO::FETCH_LAZY

PDO::FETCH_BOTH と PDO::FETCH_OBJ を結合し、アクセス時のオブジェクト変数名を作成します。

PDO::FETCH_NUM

結果セット内に戻される際の列番号 (列 0 から始まる) によって順に番号付けされた配列を戻します。

PDO::FETCH_OBJ

結果セット内に戻される列名に対応するプロパティ名を有する無名オブジェクトを戻します。

(オプション): PDO::query() または PDOStatement::execute() を呼び出したときに両方向スクロール・カーソルを指定している場合には、PDOStatement::fetch() にさらに 2 つの引数を渡すことができます。

1. このフェッチ要求のフェッチ・オリエンテーションは、次のとおりです。

PDO::FETCH_ORI_NEXT (デフォルト)

結果セット内の次の行を取り出します。

PDO::FETCH_ORI_PRIOR

結果セット内の直前の行を取り出します。

PDO::FETCH_ORI_FIRST

結果セット内の最初の行を取り出します。

PDO::FETCH_ORI_LAST

結果セット内の最後の行を取り出します。

PDO::FETCH_ORI_ABS

結果セット内の絶対行を取り出します。PDOStatement::fetch() の 3 番目の引数として正の整数が必要です。

PDO::FETCH_ORI_REL

結果セット内の相対行を取り出します。PDOStatement::fetch() の 3 番目の引数として正または負の整数が必要です。

2. 結果セット内の絶対または相対行を要求する整数。PDOStatement::fetch() の 2 番目の引数で要求されたフェッチ・オリエンテーションに対応します。

結果セット内の最後の行がすでに前方スクロール結果セット用に取得されている場合、PDOStatement::fetch() は FALSE を返します。

```
$stmt = $conn->query("SELECT firstnme, lastname FROM employee");
while ($row = $stmt->fetch(PDO::FETCH_NUM)) {
    print "Name: <p>{$row[0]} {$row[1]}</p>";
}
```

PHP でのラージ・オブジェクトのフェッチ (PDO)

PDO 拡張モジュールはすべての DB2 データ・タイプをサポートします。これには、文字ラージ・オブジェクト (CLOB) およびバイナリー・ラージ・オブジェクト (BLOB) データ・タイプが含まれます。ラージ・オブジェクトを結果セットから取り出すと、そのラージ・オブジェクトを単に PHP ストリングとして扱うことができます。しかし、ラージ・オブジェクトを PHP ストリングとして扱う方法では、PHP サーバー上で必要以上にリソースを消費することになります。ラージ・オブジェクトのデータを含むファイルを作成することが最終的な目標である場合、ラージ・オブジェクトを PHP サーバー上のファイルに直接取り出すことにより、システム・リソースを節約することができます。

システム上に PHP 5.1 以上の環境をセットアップし、PDO および PDO_IBM 拡張モジュールを使用可能にする必要があります。

ラージ・オブジェクトをデータベースからファイルに直接取り出すには、次のようにしてください。

1. ストリームを表す PHP 変数 (例えば、fopen() への呼び出しからの戻り値) を作成します。
2. PDO::prepare() を呼び出して、SQL ステートメントから PDOStatement オブジェクトを作成します。
3. PDOStatement::bindParam() を呼び出して、ラージ・オブジェクトの出力列を、ストリームを表す PHP 変数にバインドします。3 番目のパラメーターは、ラージ・オブジェクトのデータを含むファイルの名前を保持する PHP 変数の名前を表すストリングです。4 番目のパラメーターは PHP 定数、PDO::PARAM_LOB です。これは、データをファイルに書き込むように PDO 拡張モジュールに通知します。PDOStatement::bindParam() を呼び出して、結果セット内の各列に異なる PHP 変数を割り当てる必要があることに注意してください。
4. PDOStatement::execute() を呼び出して、SQL ステートメントを発行します。

5. PDOStatement::fetch(PDO::FETCH_BOUND) を呼び出して結果セット内の次の行を取り出し、PDOStatement::bindColumn() メソッドに関連付けた PHP 変数にその列の出力をバインドします。

```
$stmt = $conn->prepare("SELECT name, picture FROM animal_pictures");
$picture = fopen("/opt/albums/spook/grooming.jpg", "wb");
$stmt->bindColumn($stmt, 1, $nickname, PDO::PARAM_STR, 32);
$stmt->bindColumn($stmt, 2, $picture, PDO::PARAM_LOB);
$stmt->execute();
$stmt->fetch(PDO::FETCH_BOUND);
```

PHP でのトランザクションの管理 (PDO)

PDO は、デフォルトで、すべての接続を自動コミット・モードで開きます。自動コミット・モードは、スケーラビリティが必要な Web アプリケーションのパフォーマンスを阻害するロック・エスカレーションの問題を防ぐのに役立ちます。しかし、スクリプトによっては、1 つ以上の SQL ステートメントを含むトランザクションのロールバックが必要になる場合があります。PDO を使用することによって、トランザクションをきめ細かく制御することができます。

システム上に PHP 環境をセットアップし、PDO 拡張モジュールを使用可能にする必要があります。

トランザクションを開始するには、次のようにします。

1. PDO::beginTransaction() を呼び出して、新規トランザクションを開始します。
2. トランザクションが使用可能にされている接続リソースを使用して、データベース・トランザクションの有効範囲内で 1 つ以上の SQL ステートメントを発行します。
3. トランザクションをコミットまたはロールバックします。
 - トランザクションをコミットするには、PDO::commit() を呼び出します。
 - トランザクションをロールバックするには、PDO::rollback() を呼び出します。

トランザクションをコミットまたはロールバックした後に、PDO はデータベース接続を自動コミット・モードに自動的にリセットします。SQL ステートメントをトランザクション内で発行し、明示的にトランザクションをコミットまたはロールバックせずにスクリプトが終了する場合、PDO はトランザクション内で実行されたすべての作業を自動的にロールバックします。

```
$conn = new PDO('ibm:SAMPLE', 'db2inst1', 'ibmdb2', array(
    PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION));
// PDO::ERRMODE_EXCEPTION means an SQL error throws an exception
try {
    // Issue these SQL statements in a transaction within a try{} block
    $conn->beginTransaction();

    // One or more SQL statements

    $conn->commit();
}
catch (Exception $e) {
    // If something raised an exception in our transaction block of statements,
```

```
// roll back any work performed in the transaction
print '<p>Unable to complete transaction!</p>';
$conn->rollBack();
}
```

PHP でのエラーおよび警告の処理 (PDO)

データベースに接続しようとしたり、SQL ステートメントを発行しようとする、問題が発生することがあります。接続用のパスワードが正しくないか、SELECT ステートメントで参照した表が存在しないか、または SQL ステートメントの構文が無効である可能性があります。コード化は慎重に行い、PDO によって提供されるエラー処理関数を使用して、アプリケーションがスムーズに問題から回復できるようにする必要があります。

システム上に PHP 環境をセットアップし、PDO および PDO_IBM 拡張モジュールを使用可能にする必要があります。

PDO は、エラーを警告、エラー、または例外として取り扱うオプションを提供します。ただし、ユーザーが新規の PDO 接続オブジェクトを作成すると、PDO は、エラー発生時に必ず PDOException オブジェクトをスローします。この例外がキャッチされないと、PHP はエラー情報のバクトレースをプリントするので、それによって、ユーザー名やパスワードといったデータベース接続の信用証明情報が露出してしまふ恐れがあります。

- PDOException オブジェクトをキャッチし、関連したエラーを処理するには以下のようになります。
 1. PDO コンストラクターへの呼び出しを try ブロックに入れます。
 2. try ブロックの後に、PDOException オブジェクトをキャッチする catch ブロックを含めます。
 3. エラーに関連したエラー・メッセージを、PDOException オブジェクトの Exception::getMessage() メソッドを呼び出すことによって取得します。
- PDO または PDOStatement オブジェクトに関連した SQLSTATE を取得するには、そのオブジェクトの errorCode() メソッドを呼び出します。
- PDO または PDOStatement オブジェクトに関連したエラー情報の配列を取得するには、そのオブジェクトの errorInfo() メソッドを呼び出します。配列には、最初の要素として SQLSTATE を表す文字列が、2 つ目の要素として SQL または CLI エラー・コードを表す整数が、そして 3 つ目の要素としてフルテキストのエラー・メッセージの入った文字列が含まれます。

ストアド・プロシージャの呼び出し

PHP での、OUT または INOUT パラメーターを持つストアド・プロシージャの呼び出し (PDO)

DB2 は、入力値を受け入れるだけのパラメーター (IN パラメーター)、出力値を戻すだけのパラメーター (OUT パラメーター)、または入力値を受け入れて出力値を戻すパラメーター (INOUT) を持つストアド・プロシージャをサポートします。

PHP 用の PDO_IBM 拡張モジュールを使用すると、SQL ステートメント内のその

他のパラメーター・マーカースと同様に IN パラメーターを処理できます。しかし、PDO_IBM 拡張モジュールでは、OUT および INOUT パラメーターを持つストアド・プロシージャを呼び出し (CALL)、それらのパラメーターから出力値を取得することもできます。

システム上に PHP 環境をセットアップし、PDO および PDO_IBM 拡張モジュールを使用可能にする必要があります。

OUT または INOUT パラメーターを持つストアド・プロシージャを呼び出すには、次のようにしてください。

1. PDO::prepare() を呼び出して、OUT および INOUT パラメーターを表すパラメーター・マーカースを持つ CALL ステートメントを準備します。
2. PDOStatement::bindParam() を呼び出して、CALL ステートメントが発行された後にパラメーターの出力値を保持する PHP 変数の名前にそれぞれのパラメーター・マーカースをバインドします。INOUT パラメーターの場合、PHP 変数の値は、CALL ステートメントが発行されるときにパラメーターの入力値として渡されます。PDOStatement::bindParam() の 3 番目のパラメーターを、バインドする予定のデータのタイプに設定します。

PDO::PARAM_NULL

SQL NULL データ・タイプを表します。

PDO::PARAM_INT

SQL 整数タイプを表します。

PDO::PARAM_LOB

SQL ラージ・オブジェクト・タイプを表します。

PDO::PARAM_STR

SQL 文字データ・タイプを表します。

3. INOUT パラメーターの場合、ビット単位 OR 演算子を使用して、PDO::PARAM_INPUT_OUTPUT を、バインドする予定のデータのタイプに付加します。
4. PDOStatement::bindParam() の 4 番目のパラメーターを、出力値の期待される最大長に設定します。

```
$sql = 'CALL match_animal(?, ?)';
$stmt = $conn->prepare($sql);

$second_name = "Rickety Ride";
$weight = 0;

$stmt->bindParam(1, $second_name, PDO::PARAM_STR|PDO::PARAM_INPUT_OUTPUT, 32);
$stmt->bindParam(2, $weight, PDO::PARAM_INT, 10);

print "Values of bound parameters _before_ CALL:¥n";
print " 1: {$second_name} 2: {$weight}¥n";

$stmt->execute();

print "Values of bound parameters _after_ CALL:¥n";
print " 1: {$second_name} 2: {$weight}¥n";
```

PHP での、複数の結果セットを戻すストアード・プロシージャの呼び出し (PDO)

DB2 では、複数の結果セットを戻すストアード・プロシージャの作成および呼び出しを行うことができます。PHP 用の PDO_IBM 拡張モジュールは、`nextRowset()` メソッドによって、この機能をサポートします。このメソッドを使用して、同じストアード・プロシージャに対する単一の呼び出しで戻されるさまざまな結果セットから、行を取り出すことができます。

- システム上に PHP 5.1 以上の環境をセットアップし、PDO_IBM 拡張モジュールを使用可能にする必要があります。
- `PDO::query()` または `PDOStatement::execute()` を使用してストアード・プロシージャを呼び出すことによって戻される `PDOStatement` オブジェクトが必要です。

ストアード・プロシージャから複数の結果セットを戻すには、次のようにしてください。

1. 最初の結果セットは、CALL ステートメントによって戻される `PDOStatement` オブジェクトに関連付けられます。最初の結果セットから取り出せる行がなくなるまで、`PDOStatement` オブジェクトから行を取り出すことができます。
2. `PDOStatement` オブジェクトの `nextRowset()` メソッドを呼び出して、次の結果セットを戻します。次の結果セットから取り出せる行がなくなるまで、`PDOStatement` オブジェクトから行を取り出すことができます。

```
$sql = 'CALL multiple_results()';
$stmt = $conn->query($sql);
do {
    $rows = $stmt->fetchAll(PDO::FETCH_NUM);
    if ($rows) {
        print_r($rows);
    }
} while ($stmt->nextRowset());
```

第 2 部 Perl アプリケーションの開発

第 5 章 Perl でのプログラミングに関する考慮事項

Perl は、数多くのオペレーティング・システムで自由に使用できるポピュラーなプログラム言語です。Perl データベース・インターフェース (DBI) モジュール (<http://www.perl.com> から利用可能) で DBD::DB2 ドライバー (<http://www.ibm.com/software/data/db2/perl> から利用可能) を使用すると、Perl を使った DB2 アプリケーションを作成することができます。

Perl はインタープリター言語であり、Perl DBI モジュールは動的 SQL を使用するため、DB2 アプリケーションのプロトタイプを素早く作成および修正する上で、Perl は理想的な言語です。Perl DBI モジュールは、CLI および JDBC と大変よく似たインターフェースを使用するため、Perl プロトタイプを簡単に CLI および JDBC に移植することができます。

大部分のデータベース・ベンダーは、Perl DBI モジュールのデータベース・ドライバを提供しています。これは、さまざまなデータベース・サーバーからデータにアクセスするアプリケーションを作成するために、Perl を使用することも意味します。たとえば、DBD::Oracle データベース・ドライバを使用して Oracle データベースに接続する Perl DB2 アプリケーションを作成し、Oracle データベースからデータを取り出し、DBD::DB2 データベース・ドライバを使用して DB2 データベースにデータを挿入することができます。

Perl でのデータベース接続

Perl が DBI モジュールをロードできるようにするには、DB2 アプリケーションに次の行を含める必要があります。

```
use DBI;
```

DBI モジュールは、DBI->connect ステートメントを使用してデータベース・ハンドルを作成すると、DBD::DB2 ドライバを自動的にロードします。以下の構文を使用します。

```
my $dbh = DBI->connect('dbi:DB2:dbalias', $userID, $password);
```

詳細は次のとおりです。

\$dbh

connect ステートメントが戻すデータベース・ハンドル。

dbalias

DB2 データベース・ディレクトリーにカタログされている DB2 別名

\$userID

データベースへの接続で使用するユーザー ID

\$password

データベースへの接続で使用するユーザー ID のパスワード

Perl での取り出し結果

Perl DBI モジュールは動的 SQL しかサポートしていないため、Perl DB2 アプリケーションではホスト変数は使用できません。

SQL 照会から結果を戻すには、以下のステップを実行します。

1. DBI->connect ステートメントを使用してデータベースに接続することにより、データベース・ハンドルを作成します。
2. 作成したデータベース・ハンドルからステートメント・ハンドルを作成します。たとえば、ストリング引数として SQL ステートメントと一緒に prepare を呼び出し、データベース・ハンドルからステートメント・ハンドル *\$sth* を戻すことができます。以下に示す Perl ステートメントは、その例です。

```
my $sth = $dbh->prepare
    'SELECT firstnme, lastname
     FROM employee '
    );
```

3. ステートメント・ハンドルで execute を呼び出して、SQL ステートメントを実行します。execute が正常に呼び出されると、結果セットがステートメント・ハンドルに関連付けられます。たとえば、以下の Perl ステートメントを使用すると、前の例で準備したステートメントを実行できます。

```
#Note: $rc represents the return code for the execute call
my $rc = $sth->execute);
```

4. fetchrow) への呼び出しを使用して、ステートメント・ハンドルに関連付けられた結果セットから行を取り出します。Perl DBI は、列ごとに値を 1 つ指定した配列として行を戻します。たとえば、以下の Perl ステートメントを使用すると、前の例にあるステートメント・ハンドルからすべての行を戻すことができます。

```
while ($firstnme, $lastname) = $sth->fetchrow))
    print "$firstnme $lastname\n";
}
```

Perl のパラメーター・マーカー

指定したフィールドごとに別々の入力値を使用して、準備したステートメントを実行できるようにするため、Perl DBI モジュールはパラメーター・マーカーを使ってステートメントを準備し、実行します。SQL ステートメントにパラメーター・マーカーを入れるには、疑問符 (?) 文字を使用します。

以下の Perl コードは、SELECT ステートメントの WHERE 節のパラメーター・マーカーを受け入れるステートメント・ハンドルを作成します。その後、このコードは、入力値 25000 および 35000 を使用して 2 度ステートメントを実行し、パラメーター・マーカーを置換します。

```
my $sth = $dbh->prepare
    'SELECT firstnme, lastname
     FROM employee
     WHERE salary > ?'
    );

my $rc = $sth->execute(25000);
```

```
⋮
my $rc = $sth->execute(35000);
```

Perl の SQLSTATE および SQLCODE 変数

Perl DBI のデータベース・ハンドルまたはステートメント・ハンドルに関連付けられた SQLSTATE を戻すには、`state` メソッドを呼び出します。たとえば、データベース・ハンドル `$dbhhandle` に関連付けられた SQLSTATE を戻すには、次の Perl ステートメントをアプリケーションに組み込みます。

```
my $sqlstate = $dbhhandle->state;
```

Perl DBI のデータベース・ハンドルまたはステートメント・ハンドルに関連付けられた SQLCODE を戻すには、`err` メソッドを呼び出します。Perl DBI のデータベース・ハンドルまたはステートメント・ハンドルに関連付けられた SQLCODE のメッセージを戻すには、`errstr` メソッドを呼び出します。たとえば、データベース・ハンドル `$dbhhandle` に関連付けられた SQLCODE を戻すには、次の Perl ステートメントをアプリケーションに組み込みます。

```
my $sqlcode = $dbhhandle->err;
```

Perl の制約事項

Perl DBI モジュールがサポートするのは、動的 SQL だけです。複数回ステートメントを実行する必要がある場合には、ステートメントを準備する `prepare` 呼び出しを発行して、Perl DB2 アプリケーションのパフォーマンスを改善することができます。

Perl ではマルチスレッド・データベース・アクセスはサポートされていません。

ワークステーションにインストールする DBD::DB2 ドライバー・バージョンの制限に関する最新情報については、DBD::DB2 パッケージにある CAVEATS ファイルを参照してください。

Perl プログラムの例

以下に、Perl で作成されたアプリケーションの例を示します。

```
#!/usr/bin/perl
use DBI;

my $database='dbi:DB2:sample';
my $user='';
my $password='';

my $dbh = DBI->connect($database, $user, $password)
    or die "Can't connect to $database: $DBI::errstr";

my $sth = $dbh->prepare
    q SELECT firstme, lastname
      FROM employee }
    )
    or die "Can't prepare statement: $DBI::errstr";

my $rc = $sth->execute
    or die "Can't execute statement: $DBI::errstr";
```

```

print "Query will return ${sth->NUM_OF_FIELDS} fields.¥n¥n";
print "${sth->NAME}->0]: ${sth->NAME}->1]¥n";

while ($firstme, $lastname) = $sth->fetchrow))
    print "$firstme: $lastname¥n";
}

# check for problems which may have terminated the fetch early
warn $DBI::errstr if $DBI::err;

$sth->finish;
$dbh->disconnect;

```

Perl アプリケーションの構築

DB2 は、Perl 5.8 以降で書かれたクライアント・アプリケーションのデータベース・アクセスをサポートしています。本書の発刊時点では、Perl Database Interface (Perl DBI) バージョン 1.41 以降用の DB2 ドライバー (DBD::DB2) のリリース 1.0 が、AIX、HP-UX、Linux、Solaris および Windows でサポートされており、入手可能です。最新のドライバーの入手方法については、以下を参照してください。

<http://www.ibm.com/db2/perl>

DB2 では、Perl サンプル・プログラムが、UNIX の場合は `sqlib/samples/perl` ディレクトリーに、Windows の場合は `sqlib¥samples¥perl` ディレクトリーに用意されています。

コマンド行で Perl プログラムに対して `perl` インタープリターを実行するには、インタープリター名とプログラム名 (拡張子を含む) を入力します。

1. サーバーにローカル接続している場合:

```
perl dbauth.pl
```

2. リモート・クライアントから接続している場合:

```
perl dbauth.pl sample <userid> <password>
```

一部のプログラムでは、サポート・ファイルを実行する必要があります。 `tbsel` サンプル・プログラムでは、 `tbselcreate.db2` CLP スクリプトで複数の表を作成する必要があります。 `tbselinit` スクリプト (UNIX)、または `tbselinit.bat` バッチ・ファイル (Windows) は、 `tbseldrop.db2` を呼び出して表をドロップしてから (存在する場合)、 `tbselcreate.db2` を呼び出して表を作成します。したがって、プログラムを実行するには、次のコマンドを入力します。

1. サーバーにローカル接続している場合:

```
tbselinit
perl tbsel.pl
```

2. リモート・クライアントから接続している場合:

```
tbselinit
perl tbsel.pl sample <userid> <password>
```

注: リモート・クライアントの場合は、 `tbselinit` または `tbselinit.bat` ファイルの接続ステートメントを修正して、 `db2 connect to sample user <userid> using <password>` のように、自分のユーザー ID とパスワードをハードコーディングする必要があります。

第 3 部 付録

付録 A. DB2 技術情報の概説

DB2 技術情報は、以下のツールと方法を介して利用できます。

- DB2 インフォメーション・センター
 - トピック (タスク、概念、およびリファレンス・トピック)
 - DB2 ツールのヘルプ
 - サンプル・プログラム
 - チュートリアル
- DB2 資料
 - PDF ファイル (ダウンロード可能)
 - PDF ファイル (DB2 PDF DVD に含まれる)
 - 印刷資料
- コマンド行ヘルプ
 - コマンド・ヘルプ
 - メッセージ・ヘルプ

注: DB2 インフォメーション・センターのトピックは、PDF やハードコピー資料よりも頻繁に更新されます。最新の情報を入手するには、資料の更新が発行されたときにそれをインストールするか、ibm.com[®] にある DB2 インフォメーション・センターを参照してください。

技術資料、ホワイト・ペーパー、IBM Redbooks[®] 資料などのその他の DB2 技術情報には、オンライン (ibm.com) でアクセスできます。DB2 Information Management ソフトウェア・ライブラリー・サイト (<http://www.ibm.com/software/data/sw-library/>) にアクセスしてください。

資料についてのフィードバック

DB2 の資料についてのお客様からの貴重なご意見をお待ちしています。DB2 の資料を改善するための提案については、db2docs@ca.ibm.com まで E メールを送信してください。DB2 の資料チームは、お客様からのフィードバックすべてに目を通しますが、直接お客様に返答することはありません。お客様が関心をお持ちの内容について、可能な限り具体的な例を提供してください。特定のトピックまたはヘルプ・ファイルについてのフィードバックを提供する場合は、そのトピック・タイトルおよび URL を含めてください。

DB2 お客様サポートに連絡する場合には、この E メール・アドレスを使用しないでください。資料を参照しても、DB2 の技術的な問題が解決しない場合は、お近くの IBM サービス・センターにお問い合わせください。

DB2 テクニカル・ライブラリー (ハードコピーまたは PDF 形式)

以下の表は、DB2 ライブラリーについて説明しています。DB2 ライブラリーに関する詳細な説明については、www.ibm.com/shop/publications/order にある IBM Publications Center にアクセスしてください。英語の DB2 バージョン 9.5 のマニュアル (PDF 形式) とその翻訳版は、www.ibm.com/support/docview.wss?rs=71&uid=swg2700947 からダウンロードできます。

この表には印刷資料が入手可能かどうかを示されていますが、国または地域によっては入手できない場合があります。

表 1. DB2 の技術情報

資料名	資料番号	印刷資料が入手可能かどうか
管理 API リファレンス	SC88-4431-00	入手可能
管理ルーチンおよびビュー	SC88-4435-00	入手不可
コール・レベル・インターフェース ガイドおよびリファレンス 第 1 巻	SC88-4433-00	入手可能
コール・レベル・インターフェース ガイドおよびリファレンス 第 2 巻	SC88-4434-00	入手可能
コマンド・リファレンス	SC88-4432-00	入手可能
データ移動ユーティリティガイドおよびリファレンス	SC88-4421-00	入手可能
データ・リカバリーと高可用性ガイドおよびリファレンス	SC88-4423-00	入手可能
データ・サーバー、データベース、およびデータベース・オブジェクトのガイド	SC88-4259-00	入手可能
データベース・セキュリティ・ガイド	SC88-4418-00	入手可能
ADO.NET および OLE DB アプリケーションの開発	SC88-4425-00	入手可能
組み込み SQL アプリケーションの開発	SC88-4426-00	入手可能
Java アプリケーションの開発	SC88-4427-00	入手可能
Perl および PHP アプリケーションの開発	SC88-4428-00	入手不可
SQL および 外部ルーチンの開発	SC88-4429-00	入手可能
データベース・アプリケーション開発の基礎	GC88-4430-00	入手可能
DB2 インストールおよび管理概説 (Linux および Windows 版)	GC88-4439-00	入手可能
国際化対応ガイド	SC88-4420-00	入手可能

表 1. DB2 の技術情報 (続き)

資料名	資料番号	印刷資料が入手可能かどうか
メッセージ・リファレンス 第 1 巻	GI88-4109-00	入手不可
メッセージ・リファレンス 第 2 巻	GI88-4110-00	入手不可
マイグレーション・ガイド	GC88-4438-00	入手可能
<i>Net Search Extender</i> 管理および ユーザーズ・ガイド	SC88-4630-00	入手可能
注: この資料の内容は、DB2 インフォメーション・センターには含まれていません。		
パーティションおよびクラスタ リングのガイド	SC88-4419-00	入手可能
<i>Query Patroller</i> 管理およびユー ザーズ・ガイド	SC88-4611-00	入手可能
IBM データ・サーバー・クライ アント機能 概説およびインス トール	GC88-4441-00	入手不可
DB2 サーバー機能 概説および インストール	GC88-4440-00	入手可能
<i>Spatial Extender and Geodetic Data Management Feature</i> ユー ザーズ・ガイドおよびリファレ ンス	SC88-4629-00	入手可能
SQL リファレンス 第 1 巻	SC88-4436-00	入手可能
SQL リファレンス 第 2 巻	SC88-4437-00	入手可能
システム・モニター ガイドお よびリファレンス	SC88-4422-00	入手可能
テキスト検索ガイド	SC88-4424-00	入手可能
問題判別ガイド	GI88-4108-00	入手不可
データベース・パフォーマンス のチューニング	SC88-4417-00	入手可能
<i>Visual Explain</i> チュートリアル	SC88-4449-00	入手不可
新機能	SC88-4445-00	入手可能
ワークロード・マネージャー ガイドおよびリファレンス	SC88-4446-00	入手可能
<i>pureXML</i> ガイド	SC88-4447-00	入手可能
XQuery リファレンス	SC88-4448-00	入手不可

表 2. DB2 Connect 固有の技術情報

資料名	資料番号	印刷資料が入手可能かどうか
DB2 Connect Personal Edition 概説およびインストール	GC88-4443-00	入手可能

表 2. DB2 Connect 固有の技術情報 (続き)

資料名	資料番号	印刷資料が入手可能かどうか
DB2 Connect サーバー機能 概説およびインストール	GC88-4444-00	入手可能
DB2 Connect ユーザーズ・ガイド	SC88-4442-00	入手可能

表 3. Information Integration の技術情報

資料名	資料番号	印刷資料が入手可能かどうか
Information Integration: フェデレーテッド・システム 管理ガイド	SC88-4166-01	入手可能
Information Integration: レプリケーションおよびイベント・パブリッシングのための ASNCLP プログラム・リファレンス	SC88-4167-02	入手可能
Information Integration: フェデレーテッド・データ・ソース 構成ガイド	SC88-4185-01	入手不可
Information Integration: SQL レプリケーション ガイドおよびリファレンス	SC88-4168-01	入手可能
Information Integration: レプリケーションとイベント・パブリッシング 概説	GC88-4187-01	入手可能

DB2 の印刷資料の注文方法

DB2 の印刷資料が必要な場合、オンラインで購入することができますが、すべての国および地域で購入できるわけではありません。DB2 の印刷資料については、IBM 営業担当員にお問い合わせください。DB2 PDF ドキュメンテーション DVD の一部のソフトコピー・ブックは、印刷資料では入手できないことに留意してください。例えば、「DB2 メッセージ・リファレンス」はどちらの巻も印刷資料としては入手できません。

DB2 PDF ドキュメンテーション DVD で利用できる DB2 の印刷資料の大半は、IBM に有償で注文することができます。国または地域によっては、資料を IBM Publications Center からオンラインで注文することもできます。お客様の国または地域でオンライン注文が利用できない場合、DB2 の印刷資料については、IBM 営業担当員にお問い合わせください。DB2 PDF ドキュメンテーション DVD に収録されている資料の中には、印刷資料として提供されていないものもあります。

注: 最新で完全な DB2 資料は、DB2 インフォメーション・センター (<http://publib.boulder.ibm.com/infocenter/db2luw/v9r5>) で参照することができます。

DB2 の印刷資料は以下の方法で注文することができます。

- 日本 IBM 発行のマニュアルはインターネット経由でご購入いただけます。詳しくは <http://www.ibm.com/shop/publications/order> の「ご注文について」をご覧ください。資料の注文情報にアクセスするには、お客様の国、地域、または言語を選択してください。その後、各ロケーションにおける注文についての指示に従ってください。
- DB2 の印刷資料を IBM 営業担当員に注文するには、以下のようになります。
 1. 以下の Web サイトのいずれかから、営業担当員の連絡先情報を見つけてください。
 - IBM Directory of world wide contacts (www.ibm.com/planetwide)
 - IBM Publications Web サイト (<http://www.ibm.com/shop/publications/order>)
国、地域、または言語を選択し、お客様の所在地に該当する Publications ホーム・ページにアクセスしてください。このページから、「このサイトについて」のリンクにアクセスしてください。
 2. 電話をご利用の場合は、DB2 資料の注文であることをご指定ください。
 3. 担当者に、注文する資料のタイトルと資料番号をお伝えください。タイトルと資料番号は、44 ページの『DB2 テクニカル・ライブラリー (ハードコピーまたは PDF 形式)』でご確認いただけます。

コマンド行プロセッサから SQL 状態ヘルプを表示する

DB2 は、SQL ステートメントの結果の原因になったと考えられる条件の SQLSTATE 値を戻します。SQLSTATE ヘルプは、SQL 状態および SQL 状態クラス・コードの意味を説明します。

SQL 状態ヘルプを呼び出すには、コマンド行プロセッサを開いて以下のように入力します。

```
? sqlstate or ? class code
```

ここで、*sqlstate* は有効な 5 桁の SQL 状態を、*class code* は SQL 状態の最初の 2 桁を表します。

例えば、? 08003 を指定すると SQL 状態 08003 のヘルプが表示され、? 08 を指定するとクラス・コード 08 のヘルプが表示されます。

異なるバージョンの DB2 インフォメーション・センターへのアクセス

DB2 バージョン 9.5 のトピックを扱っている DB2 インフォメーション・センターの URL は、<http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/>です。

DB2 バージョン 9 のトピックを扱っている DB2 インフォメーション・センターの URL は <http://publib.boulder.ibm.com/infocenter/db2luw/v9/>です。

DB2 バージョン 8 のトピックについては、バージョン 8 のインフォメーション・センターの URL <http://publib.boulder.ibm.com/infocenter/db2luw/v8/>にアクセスしてください。

DB2 インフォメーション・センターにおける特定の言語でのトピックの表示

DB2 インフォメーション・センターでは、ブラウザの設定で指定した言語でのトピックの表示が試みられます。トピックがその指定言語に翻訳されていない場合は、DB2 インフォメーション・センターでは英語でトピックが表示されます。

- Internet Explorer Web ブラウザーで、指定どおりの言語でトピックを表示するには、以下のようにします。
 1. Internet Explorer の「ツール」 -> 「インターネット オプション」 -> 「言語 ...」 ボタンをクリックします。「言語の優先順位」ウィンドウがオープンします。
 2. 該当する言語が、言語リストの先頭の項目に指定されていることを確認します。
 - リストに新しい言語を追加するには、「追加...」 ボタンをクリックします。
 3. ブラウザー・キャッシュを消去してから、ページをリフレッシュし、使用する言語で DB2 インフォメーション・センターを表示します。
- Firefox または Mozilla Web ブラウザーの場合に、使いたい言語でトピックを表示するには、以下のようにします。
 1. 「ツール」 -> 「オプション」 -> 「詳細」 ダイアログの「言語」セクションにあるボタンを選択します。「設定」ウィンドウに「言語」パネルが表示されます。
 2. 該当する言語が、言語リストの先頭の項目に指定されていることを確認します。
 - リストに新しい言語を追加するには、「追加...」 ボタンをクリックしてから、「言語を追加」ウィンドウで言語を選択します。
 - リストの先頭に新しい言語を移動するには、その言語を選択してから、その言語が言語リストに先頭に行くまで「上に移動」 ボタンをクリックします。
 3. ブラウザー・キャッシュを消去してから、ページをリフレッシュし、使用する言語で DB2 インフォメーション・センターを表示します。

ブラウザとオペレーティング・システムの組み合わせによっては、オペレーティング・システムの地域の設定も希望のロケールと言語に変更しなければならない場合があります。

コンピューターまたはイントラネット・サーバーにインストールされた DB2 インフォメーション・センターの更新

DB2 インフォメーション・センターをローカルにインストールしている場合は、IBM から提供される更新をダウンロードおよびインストールすることができます。

ローカルにインストールされた DB2 インフォメーション・センターを更新するには、以下のことを行う必要があります。

1. コンピューター上の DB2 インフォメーション・センターを停止し、インフォメーション・センターをスタンドアロン・モードで再始動します。インフォメーション・センターをスタンドアロン・モードで実行すると、ネットワーク上の他のユーザーがそのインフォメーション・センターにアクセスできなくなります。これで、更新をダウンロードして適用できるようになります。
2. 「更新」機能を使用することにより、どんな更新が利用できるかを確認します。インストールする更新がある場合は、「更新」機能を使用してそれをダウンロードおよびインストールできます。

注: ご使用の環境において、インターネットに接続されていないマシンに DB2 インフォメーション・センターの更新をインストールする必要がある場合は、インターネットに接続されていて DB2 インフォメーション・センターがインストールされているマシンを使用して、更新サイトをローカル・ファイル・システムにミラーリングする必要があります。ネットワーク上の多数のユーザーが資料の更新をインストールする場合にも、更新サイトをローカルにミラーリングして、更新サイト用のプロキシーを作成することにより、個々のユーザーが更新を実行するのに要する時間を短縮できます。

更新パッケージが入手可能な場合、「更新」機能を使用してパッケージをダウンロードします。ただし、「更新」機能は、スタンドアロン・モードでのみ使用できます。

3. スタンドアロンのインフォメーション・センターを停止し、コンピューター上の DB2 インフォメーション・センターを再開します。

注: Windows Vista の場合、下記のコマンドは管理者として実行する必要があります。完全な管理者特権でコマンド・プロンプトまたはグラフィカル・ツールを起動するには、ショートカットを右クリックしてから、「管理者として実行」を選択します。

コンピューターまたはイントラネット・サーバーにインストール済みの DB2 インフォメーション・センターを更新するには、以下のようになります。

1. DB2 インフォメーション・センターを停止します。
 - Windows では、「スタート」→「コントロール パネル」→「管理ツール」→「サービス」をクリックします。次に、「DB2 インフォメーション・センター」サービスを右クリックして「停止」を選択します。
 - Linux では、以下のコマンドを入力します。

```
/etc/init.d/db2icdv95 stop
```
2. インフォメーション・センターをスタンドアロン・モードで開始します。
 - Windows の場合:
 - a. コマンド・ウィンドウを開きます。

- b. インフォメーション・センターがインストールされているパスにナビゲートします。デフォルトでは、DB2 インフォメーション・センターは <Program Files>¥IBM¥DB2 Information Center¥Version 9.5 ディレクトリーにインストールされています (<Program Files> は「Program Files」ディレクトリーのロケーション)。
- c. インストール・ディレクトリーの doc¥bin ディレクトリーにナビゲートします。
- d. 次のように help_start.bat ファイルを実行します。

```
help_start.bat
```

- Linux の場合:

- a. インフォメーション・センターがインストールされているパスにナビゲートします。デフォルトでは、DB2 インフォメーション・センターは /opt/ibm/db2ic/V9.5 ディレクトリーにインストールされています。
- b. インストール・ディレクトリーの doc/bin ディレクトリーにナビゲートします。
- c. 次のように help_start スクリプトを実行します。

```
help_start
```

システムのデフォルト Web ブラウザーが起動し、スタンドアロンのインフォメーション・センターが表示されます。

3. 「更新」ボタン (🔄) をクリックします。インフォメーション・センターの右側のパネルで、「更新の検索 (Find Updates)」をクリックします。既存の文書に対する更新のリストが表示されます。
4. ダウンロード・プロセスを開始するには、ダウンロードする更新をチェックして選択し、「更新のインストール (Install Updates)」をクリックします。
5. ダウンロードおよびインストール・プロセスが完了したら、「完了」をクリックします。
6. スタンドアロンのインフォメーション・センターを停止します。

- Windows の場合は、インストール・ディレクトリーの doc¥bin ディレクトリーにナビゲートしてから、次のように help_end.bat ファイルを実行します。

```
help_end.bat
```

注: help_end バッチ・ファイルには、help_start バッチ・ファイルを使用して開始したプロセスを安全に終了するのに必要なコマンドが含まれています。Ctrl-C または他の方法を使用して、help_start.bat を終了しないでください。

- Linux の場合は、インストール・ディレクトリーの doc/bin ディレクトリーにナビゲートしてから、次のように help_end スクリプトを実行します。

```
help_end
```

注: help_end スクリプトには、help_start スクリプトを使用して開始したプロセスを安全に終了するのに必要なコマンドが含まれています。他の方法を使用して、help_start スクリプトを終了しないでください。

7. DB2 インフォメーション・センターを再開します。

- Windows では、「スタート」→「コントロール パネル」→「管理ツール」→「サービス」をクリックします。次に、「DB2 インフォメーション・センター」サービスを右クリックして「開始」を選択します。
- Linux では、以下のコマンドを入力します。

```
/etc/init.d/db2icdv95 start
```

更新された DB2 インフォメーション・センターに、更新された新しいトピックが表示されます。

DB2 チュートリアル

DB2 チュートリアルは、DB2 製品のさまざまな機能について学習するのを支援します。この演習をとおして段階的に学習することができます。

はじめに

インフォメーション・センター (<http://publib.boulder.ibm.com/infocenter/db2help/>) から、このチュートリアルの XHTML 版を表示できます。

演習の中で、サンプル・データまたはサンプル・コードを使用する場合があります。個々のタスクの前提条件については、チュートリアルを参照してください。

DB2 チュートリアル

チュートリアルを表示するには、タイトルをクリックします。

「*pureXML* ガイド」の『**pureXML™**』

XML データを保管し、ネイティブ XML データ・ストアに対して基本的な操作を実行できるように、DB2 データベースをセットアップします。

「*Visual Explain* チュートリアル」の『**Visual Explain**』

Visual Explain を使用して、パフォーマンスを向上させるために SQL ステートメントを分析し、最適化し、調整します。

DB2 トラブルシューティング情報

DB2 製品を使用する際に役立つ、トラブルシューティングおよび問題判別に関する広範囲な情報を利用できます。

DB2 ドキュメンテーション

トラブルシューティング情報は、DB2 問題判別ガイド、または DB2 インフォメーション・センターの「サポートおよびトラブルシューティング」セクションにあります。ここでは、DB2 診断ツールおよびユーティリティーを使用して、問題を切り分けて識別する方法、最も頻繁に起こる幾つかの問題に対するソリューションについての情報、および DB2 製品を使用する際に発生する可能性のある問題の解決方法についての他のアドバイスがあります。

DB2 Technical Support の Web サイト

現在問題が発生していて、考えられる原因とソリューションを検索したい場合は、DB2 Technical Support の Web サイトを参照してください。

Technical Support サイトには、最新の DB2 資料、TechNotes、プログラム

診断依頼書 (APAR またはバグ修正)、フィックスパック、およびその他のリソースへのリンクが用意されています。この知識ベースを活用して、問題に対する有効なソリューションを探し出すことができます。

DB2 Technical Support の Web サイト (<http://www.ibm.com/software/data/db2/udb/support.html>) にアクセスしてください。

ご利用条件

これらの資料は、以下の条件に同意していただける場合に限りご使用いただけます。

個人使用: これらの資料は、すべての著作権表示その他の所有権表示をしていただくことを条件に、非商業的な個人による使用目的に限り複製することができます。ただし、IBM の明示的な承諾をえずに、これらの資料またはその一部について、二次的著作物を作成したり、配布 (頒布、送信を含む) または表示 (上映を含む) することはできません。

商業的使用: これらの資料は、すべての著作権表示その他の所有権表示をしていただくことを条件に、お客様の企業内に限り、複製、配布、および表示することができます。ただし、IBM の明示的な承諾をえずにこれらの資料の二次的著作物を作成したり、お客様の企業外で資料またはその一部を複製、配布、または表示することはできません。

ここで明示的に許可されているもの以外に、資料や資料内に含まれる情報、データ、ソフトウェア、またはその他の知的所有権に対するいかなる許可、ライセンス、または権利を明示的にも黙示的にも付与するものではありません。

資料の使用が IBM の利益を損なうと判断された場合や、上記の条件が適切に守られていないと判断された場合、IBM はいつでも自らの判断により、ここで与えた許可を撤回できるものとさせていただきます。

お客様がこの情報をダウンロード、輸出、または再輸出する際には、米国のすべての輸出入関連法規を含む、すべての関連法規を遵守するものとします。

IBM は、これらの資料の内容についていかなる保証もしません。これらの資料は、特定物として現存するままの状態を提供され、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任なしで提供されます。

付録 B. 特記事項

本書は米国 IBM が提供する製品およびサービスについて作成したものです。

本書に記載の製品、サービス、または機能が日本においては提供されていない場合があります。日本で利用可能な製品、サービス、および機能については、日本 IBM の営業担当員にお尋ねください。本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、またはサービスのみが使用可能であることを意味するものではありません。これらに代えて、IBM の知的所有権を侵害することのない、機能的に同等の製品、プログラム、またはサービスを使用することができます。ただし、IBM 以外の製品とプログラムの操作またはサービスの評価および検証は、お客様の責任で行っていただきます。

IBM は、本書に記載されている内容に関して特許権 (特許出願中のものを含む) を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。実施権についてのお問い合わせは、書面にて下記宛先にお送りください。

〒106-8711
東京都港区六本木 3-2-12
日本アイ・ビー・エム株式会社
法務・知的財産
知的財産権ライセンス渉外

以下の保証は、国または地域の法律に沿わない場合は、適用されません。 IBM およびその直接または間接の子会社は、本書を特定物として現存するままの状態を提供し、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。

この情報には、技術的に不適切な記述や誤植を含む場合があります。本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

本書は、IBM 以外の Web サイトおよびリソースへのリンクまたは参照を含む場合があります。IBM は、本書より参照もしくはアクセスできる、または本書からリンクされた IBM 以外の Web サイトもしくは第三者のリソースに対して一切の責任を負いません。IBM 以外の Web サイトにリンクが張られていることにより IBM が当該 Web サイトを推奨するものではなく、またその内容、使用もしくはサイトの所有者について IBM が責任を負うことを意味するものではありません。また、IBM は、お客様が IBM Web サイトから第三者の存在を知ることになった場合にも (もしくは、IBM Web サイトから第三者へのリンクを使用した場合にも)、お客様と第三者との間のいかなる取引に対しても一切責任を負いません。従って、お客様は、IBM が上記の外部サイトまたはリソースの利用について責任を負うものではなく、また、外部サイトまたはリソースからアクセス可能なコンテンツ、サービス、

製品、またはその他の資料一切に対して IBM が責任を負うものではないことを承諾し、同意するものとします。第三者により提供されるソフトウェアには、そのソフトウェアと共に提供される固有の使用条件が適用されます。

IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。

本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム (本プログラムを含む) との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。

IBM Canada Limited
Office of the Lab Director
8200 Warden Avenue
Markham, Ontario
L6G 1C7
CANADA

本プログラムに関する上記の情報は、適切な使用条件の下で使用することができますが、有償の場合もあります。

本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、IBM 所定のプログラム契約の契約条項、IBM プログラムのご使用条件、またはそれと同等の条項に基づいて、IBM より提供されます。

この文書に含まれるいかなるパフォーマンス・データも、管理環境下で決定されたものです。そのため、他の操作環境で得られた結果は、異なる可能性があります。一部の測定が、開発レベルのシステムで行われた可能性がありますが、その測定値が、一般に利用可能なシステムのものと同じである保証はありません。さらに、一部の測定値が、推定値である可能性があります。実際の結果は、異なる可能性があります。お客様は、お客様の特定の環境に適したデータを確かめる必要があります。

IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。IBM は、それらの製品のテストは行っておりません。したがって、他社製品に関する実行性、互換性、またはその他の要求については確認できません。IBM 以外の製品の性能に関する質問は、それらの製品の供給者をお願いします。

IBM の将来の方向または意向に関する記述については、予告なしに変更または撤回される場合があります、単に目標を示しているものです。

本書には、日常の業務処理で用いられるデータや報告書の例が含まれています。より具体性を与えるために、それらの例には、個人、企業、ブランド、あるいは製品などの名前が含まれている場合があります。これらの名称はすべて架空のものであり、名称や住所が類似する企業が実在しているとしても、それは偶然にすぎません。

著作権使用許諾:

本書には、様々なオペレーティング・プラットフォームでのプログラミング手法を例示するサンプル・アプリケーション・プログラムがソース言語で掲載されています。お客様は、サンプル・プログラムが書かれているオペレーティング・プラットフォームのアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。このサンプル・プログラムは、あらゆる条件下における完全なテストを経ていません。従って IBM は、これらのサンプル・プログラムについて信頼性、利便性もしくは機能性があることをほのめかしたり、保証することはできません。

それぞれの複製物、サンプル・プログラムのいかなる部分、またはすべての派生した創作物には、次のように、著作権表示を入れていただく必要があります。

© (お客様の会社名) (西暦年). このコードの一部は、IBM Corp. のサンプル・プログラムから取られています。 © Copyright IBM Corp. _年を入れる_. All rights reserved.

商標

DB2 バージョン 9.5 ドキュメンテーション・ライブラリーの資料に記載されている会社名、製品名、またはサービス名は、IBM Corporation の商標である可能性があります。IBM Corporation の商標については、<http://www.ibm.com/legal/copytrade.shtml> を参照してください。

以下は、それぞれ各社の商標または登録商標です。

Microsoft[®]、Windows、Windows NT[®]、および Windows ロゴは、Microsoft Corporation の米国およびその他の国における商標です。

Intel[®]、Intel ロゴ、Intel Inside[®] ロゴ、Intel Centrino[®]、Intel Centrino ロゴ、Celeron[®]、Intel Xeon[®]、Intel SpeedStep[®]、Itanium[®] および Pentium[®] は、Intel Corporation の米国およびその他の国における商標です。

Java[™] およびすべての Java 関連の商標およびロゴは、Sun Microsystems, Inc. の米国およびその他の国における商標です。

UNIX は、The Open Group の米国およびその他の国における登録商標です。

Linux は、Linus Torvalds の米国およびその他の国における商標です。

Adobe[®]、Adobe ロゴ、PostScript[®]、PostScript ロゴは、Adobe Systems Incorporated の米国およびその他の国における登録商標または商標です。

他の会社名、製品名およびサービス名等はそれぞれ各社の商標です。

索引

日本語, 数字, 英字, 特殊文字の順に配列されています。なお, 濁音と半濁音は清音と同等に扱われています。

[ア行]

アプリケーション開発

Perl

アプリケーションの構築 40

アプリケーション設計

Perl でのプロトタイプリング 37

Perl の例 39

印刷資料

注文 46

インフォメーション・センター

更新 49

バージョン 47

別の言語で表示する 48

エラー処理

Perl 39

[カ行]

更新

インフォメーション・センター 49

DB2 インフォメーション・センター 49

ご利用条件

資料の使用 52

[サ行]

資料

使用に関するご利用条件 52

PDF および印刷資料 44

資料の概説 43

静的 SQL

Perl、サポートされていない 39

[タ行]

チュートリアル

トラブルシューティングと問題判別 51

Visual Explain 51

データの検索

Perl 38

データベース

Perl での接続 37

動的 SQL

Perl サポート 37

特記事項 53

トラブルシューティング

オンライン情報 51

チュートリアル 51

[ハ行]

パラメーター・マーカー

Perl 38

ヘルプ

表示 48

SQL ステートメントの 47

ホスト変数

Perl ではサポートされていない 38

[マ行]

問題判別

オンライン情報 51

チュートリアル 51

[ラ行]

例

Perl プログラム 39

D

DB2 インフォメーション・センター

更新 49

バージョン 47

別の言語で表示する 48

DB2 資料の印刷方法 46

P

Perl

アプリケーションの構築 40

アプリケーションの例 39

制限 39

データの戻り 38

データベースへの接続 37

ドライバー 37

パラメーター・マーカー 38

プログラミングについての考慮事項 37

SQLCODE 39

SQLSTATE 39

PHP 3

概要 3

PHP (続き)

セットアップ

Linux 6

Windows 5

ibm_db2

エラーおよび警告 19

結果行の取り出し 17

結果列のフェッチ 16

ストアド・プロシージャの結果 21

ストアド・プロシージャのパラメーター 20

データベースへの接続 9

データベース・メタデータの取得 10

トランザクション 18

ラージ・オブジェクトの挿入 15

ラージ・オブジェクトのフェッチ 17

SQL ステートメントの実行 12

SQL ステートメントの準備 13

XQuery 式 11

pdo_ibm

エラーおよび警告 32

結果行の取り出し 28

結果列のフェッチ 28

結果を戻さない SQL ステートメントの実行 24

結果を戻す SQL ステートメントの実行 25

ストアド・プロシージャの結果 34

ストアド・プロシージャのパラメーター 32

データベースへの接続 23

トランザクション 31

ラージ・オブジェクトの挿入 27

ラージ・オブジェクトのフェッチ 30

SQL ステートメントの準備 25

S

SQL ステートメント

ヘルプを表示する 47

V

Visual Explain

チュートリアル 51



Printed in Japan

SC88-4428-00



日本アイ・ビー・エム株式会社
〒106-8711 東京都港区六本木3-2-12

Spine information:

DB2 Version 9.5 for Linux, UNIX, and Windows

Perl および PHP アプリケーションの開発

