



**Desarrollo de aplicaciones Perl y PHP**  
**Actualizado en marzo de 2008**





**Desarrollo de aplicaciones Perl y PHP**  
**Actualizado en marzo de 2008**

**Nota**

Antes de utilizar esta información y el producto al que da soporte, lea la información general contenida en el apartado Apéndice B, "Avisos", en la página 53.

**Nota de edición**

Esta publicación es la traducción del original inglés: DB2 Version 9.5 for Linux, UNIX, and Windows - Developing Perl and PHP Applications, (SC23-5854-01).

Este documento contiene información propiedad de IBM. Se proporciona según un acuerdo de licencia y está protegido por la ley de la propiedad intelectual. La información contenida en esta publicación no incluye ninguna garantía de producto, por lo que ninguna declaración proporcionada en este manual deberá interpretarse como tal.

Puede realizar pedidos de publicaciones de IBM en línea o a través del representante de IBM de su localidad.

- Para realizar pedidos en línea, vaya a IBM Publications Center ubicado en el sitio web [www.ibm.com/shop/publications/order](http://www.ibm.com/shop/publications/order)
- Para encontrar al representante de IBM de su localidad, vaya al IBM Directory of Worldwide Contacts en el sitio web [www.ibm.com/planetwide](http://www.ibm.com/planetwide)

Para realizar pedidos de publicaciones de DB2 desde DB2 Marketing and Sales, en los EE.UU. o en Canadá, llame al 1-800-IBM-4YOU (426-4968).

Cuando envía información a IBM, otorga a IBM un derecho no exclusivo a utilizar o distribuir dicha información en la forma en que IBM considere adecuada, sin contraer por ello ninguna obligación con el remitente.

© Copyright International Business Machines Corporation 2006, 2008. Reservados todos los derechos.

# Contenido

## Parte 1. Desarrollo de aplicaciones PHP . . . . . 1

### Capítulo 1. Introducción al desarrollo de aplicaciones PHP para DB2 . . . . . 3

### Capítulo 2. Configuración del entorno PHP . . . . . 5

Configuración del entorno de PHP en Windows . . . 5  
Configuración del entorno PHP en Linux o UNIX . . . 5

### Capítulo 3. Desarrollo con ibm\_db2 . . . . . 9

Conexión a una base de datos DB2 con PHP (ibm\_db2) . . . . . 9

Recuperación de metadatos de la base de datos(ibm\_db2) . . . . . 9

Ejecución de expresiones XQuery en PHP (ibm\_db2) 11

Ejecución de sentencias de SQL. . . . . 12

Ejecución de una única sentencia SQL en PHP (ibm\_db2) . . . . . 12

Preparación y ejecución de sentencias SQL en PHP (ibm\_db2) . . . . . 13

Inserción de objetos grandes en PHP (ibm\_db2) 14

Lectura de conjuntos de resultados de consultas . . . 15

Captación de columnas del conjunto de resultados en PHP (ibm\_db2) . . . . . 15

Captación de filas del conjunto de resultados en PHP (ibm\_db2) . . . . . 16

Captación de objetos grandes en PHP (ibm\_db2) 16

Gestión de transacciones en PHP (ibm\_db2) . . . 17

Gestión de errores y mensajes de aviso (ibm\_db2) 18

Invocación de procedimientos almacenados. . . . . 19

Llamada a procedimientos almacenados con parámetros OUT o INOUT en PHP (ibm\_db2). . . 19

Llamada a procedimientos almacenados que devuelven varios conjuntos de resultados en PHP (ibm\_db2) . . . . . 20

### Capítulo 4. Desarrollo con PDO . . . . . 23

Conexión a una base de datos DB2 con PHP (PDO) 23

Ejecución de sentencias de SQL. . . . . 24

Ejecución de una única sentencia SQL en PHP que no devuelve ningún conjunto de resultados (PDO) . . . . . 24

Ejecución de una única sentencia SQL en PHP que devuelve un conjunto de resultados (PDO) . 24

Preparación y ejecución de sentencias SQL (PDO) 25

Inserción de objetos grandes en PHP (PDO) . . . 26

Lectura de conjuntos de resultados de consultas . . . 27

Captación de columnas de conjuntos de resultados en PHP (PDO). . . . . 27

Captación de filas de conjuntos de resultados en PHP (PDO) . . . . . 28

Captación de objetos grandes en PHP (PDO) . . . 29

Gestión de transacciones en PHP (PDO). . . . . 30

Manejo de errores y avisos en PHP (PDO) . . . . . 31

Invocación de procedimientos almacenados. . . . . 31

Llamada a procedimientos almacenados con parámetros OUT o INOUT en PHP (PDO) . . . . 31

Llamada a procedimientos almacenados que devuelven varios conjuntos de resultados en PHP (PDO) . . . . . 32

## Parte 2. Desarrollo de aplicaciones Perl . . . . . 35

### Capítulo 5. Consideraciones sobre la programación en Perl . . . . . 37

Conexiones de bases de datos en Perl . . . . . 37

Captación de resultados en Perl . . . . . 38

Marcadores de parámetros en Perl. . . . . 38

Variables SQLSTATE y SQLCODE en Perl . . . . . 39

Restricciones de Perl . . . . . 39

Ejemplo de programa Perl . . . . . 39

Creación de aplicaciones Perl . . . . . 40

## Parte 3. Apéndices. . . . . 41

### Apéndice A. Visión general de la información técnica de DB2 . . . . . 43

Biblioteca técnica de DB2 en copia impresa o en formato PDF . . . . . 43

Pedido de manuales de DB2 en copia impresa. . . 46

Visualización de la ayuda para estados de SQL desde el procesador de línea de mandatos . . . . 47

Acceso a diferentes versiones del Centro de información de DB2 . . . . . 47

Visualización de temas en su idioma preferido en el Centro de información de DB2 . . . . . 47

Actualización del Centro de información de DB2 instalado en el sistema o en el servidor de intranet . 48

Guías de aprendizaje de DB2 . . . . . 50

Información de resolución de problemas de DB2 . . 50

Términos y condiciones . . . . . 51

### Apéndice B. Avisos . . . . . 53

### Índice . . . . . 57



---

## Parte 1. Desarrollo de aplicaciones PHP





---

## Capítulo 1. Introducción al desarrollo de aplicaciones PHP para DB2

PHP: Hypertext Preprocessor (PHP) es un lenguaje de programación interpretado principalmente pensado para el desarrollo de aplicaciones Web. La primera versión de PHP fue creada por Rasmus Lerdorf y recibió contribuciones bajo una licencia de código abierto en 1995. Inicialmente, PHP era un motor de plantillas HTML muy sencillo, pero con el tiempo los desarrolladores de PHP han ido añadiendo funciones de acceso a bases de datos, han reescrito el intérprete, han incorporado soporte orientado a objetos y han mejorado el rendimiento. Actualmente, PHP se ha convertido en un lenguaje muy utilizado para el desarrollo de aplicaciones Web porque se centra en soluciones prácticas y da soporte a las funciones más utilizadas en aplicaciones Web.

Para facilitar la instalación y configuración en los sistemas operativos Linux, UNIX o Windows puede descargar e instalar Zend Core for IBM para utilizarlo en sistemas de producción. Puede obtener soporte, con tarifa, para Zend Core for IBM de Zend. En Windows, encontrará versiones binarias precompiladas de PHP que puede descargar de <http://php.net/>. La mayoría de las distribuciones Linux incluyen una versión precompilada de PHP. En sistemas operativos UNIX que no incluyen una versión precompilada de PHP, puede compilar su propia versión de PHP.

PHP es un lenguaje modular que le permite personalizar la funcionalidad disponible mediante el uso de extensiones. Esas extensiones pueden simplificar tareas como la lectura, escritura y manipulación de XML, la creación de clientes y servidores SOAP y el cifrado de comunicaciones entre servidor y navegador. Sin embargo, las extensiones más utilizadas para PHP proporcionan acceso de lectura y escritura a bases de datos, por lo que puede crear fácilmente un sitio Web dinámico destinado a bases de datos.

Se ha añadido al soporte de PHP desarrollando una extensión nueva denominada `pdo_ibm` para todo el que desee utilizar la interfaz de objetos de aplicación PHP (PDO). Esta nueva extensión junto con la extensión `ibm_db2` existente se incluirá como parte de IBM Data Server Client. La mayor parte de las versiones actualizadas de `ibm_db2` y `pdo_ibm` están disponibles en la biblioteca PECL (PHP Extension Community Library) <http://pecl.php.net/>. Puede utilizar sus extensiones para acceder a los datos almacenados en una base de datos DB2 mediante su aplicación PHP. A continuación se describen detalladamente las diferencias entre las extensiones:

- `ibm_db2` es una extensión escrita, mantenida y soportada por IBM para el acceso a bases de datos DB2. La extensión `ibm_db2` ofrece una interfaz de programación de aplicaciones (API) de procedimientos que, además de las operaciones normales de creación, lectura, actualización y escritura de bases de datos, ofrece un amplio acceso a metadatos de bases de datos. Puede compilar la extensión `ibm_db2` con PHP 4 o con PHP 5.
- `pdo_ibm` es un controlador para la extensión de objetos de datos PHP (PDO) que ofrece acceso a las bases de datos DB2 mediante la interfaz de base de datos orientadas a objetos estándar introducida en PHP 5.1.

Una tercera extensión, Unified ODBC, ha ofrecido históricamente acceso a los sistemas de bases de datos DB2. No se recomienda escribir nuevas aplicaciones con

esta extensión porque tanto `ibm_db2` como `pdo_ibm` ofrecen ventajas significativas en cuanto a rendimiento y estabilidad en comparación con Unified ODBC. La API de la extensión `ibm_db2` convierte la tarea de transportar una aplicación escrita anteriormente para Unified ODBC en algo casi tan sencillo como realizar un cambio global del nombre de función `odbc_` por `db2_` en el código fuente de la aplicación.

---

## Capítulo 2. Configuración del entorno PHP

---

### Configuración del entorno de PHP en Windows

DB2 da soporte al acceso de bases de datos para aplicaciones cliente escritas en el lenguaje de programación PHP utilizando la extensión `ibm_db2` o el controlador `pdo_ibm` para la extensión PHP Data Objects (PDO), o bien ambos. Para instalar una versión binaria de PHP con soporte de DB2 en Windows, puede descargar e instalar el programa disponible y gratuito Zend Core para IBM de <http://zend.com/core/ibm/> . Sin embargo, también puede instalar manualmente la versión binaria precompilada de PHP en Windows.

Apache HTTP Server debe estar instalado en el sistema.

Para instalar una versión precompilada de PHP desde <http://www.php.net> y habilitar el soporte de DB2 en Windows:

1. Descargue la última versión del paquete zip de PHP y el grupo del paquete zip de módulos PECL de <http://www.php.net>. La última versión de PHP en el momento de escribir este manual es PHP 5.2.3.
2. Extraiga el paquete zip PHP en un directorio de instalación.
3. Extraiga el grupo del paquete zip de módulos PECL en el subdirectorio `\ext\` del directorio de instalación de PHP.
4. Cree un nuevo archivo denominado `php.ini` en el directorio de instalación realizando una copia del archivo `php.ini-recommended`.
5. Abra el archivo `php.ini` en un editor de texto y añada las siguientes líneas.
  - Para habilitar la extensión PDO y el controlador `pdo_ibm`:

```
extension=php_pdo.dll
extension=php_pdo_ibm.dll
```
  - Para habilitar la extensión `ibm_db2`:

```
extension=php_ibm_db2.dll
```
6. Habilite el soporte de PHP en Apache HTTP Server 2.x añadiendo las siguientes líneas al archivo `httpd.conf`, donde `phpdir` es el directorio de instalación de PHP:

```
LoadModule php5_module 'phpdir/php5apache2.dll'
AddType application/x-httpd-php .php
PHPIniDir 'phpdir'
```
7. Vuelva a iniciar Apache HTTP Server para habilitar la configuración modificada.

---

### Configuración del entorno PHP en Linux o UNIX

DB2 da soporte al acceso de bases de datos para aplicaciones cliente escritas en el lenguaje de programación PHP utilizando la extensión `ibm_db2` o el controlador `pdo_ibm` para la extensión PHP Data Objects (PDO), o bien ambos. Para instalar una versión binaria de PHP con soporte de DB2 en Linux o AIX, puede descargar e instalar el programa disponible y gratuito Zend Core para IBM de <http://zend.com/core/ibm/> . Sin embargo, también puede compilar e instalar manualmente PHP desde el programa fuente.

- Apache HTTP Server debe estar instalado en el sistema.

- Los archivos de cabecera de desarrollo y las bibliotecas de DB2 deben estar instalados en el sistema.
- El compilador gcc y otros paquetes de desarrollo que incluyen apache-devel, autoconf, automake, bison, flex, gcc y el paquete libxml2-devel deben estar instalados en el sistema.

Para compilar PHP desde el programa fuente con soporte para DB2 en Linux o UNIX:

1. Descargue la última versión de PHP tarball de <http://www.php.net>. La última versión de PHP en el momento de escribir este manual es PHP 5.2.3.
2. Desempaque el archivo con el siguiente mandato:

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

3. Cambie el directorio al directorio php-5.x.x recién creado.
4. Configure el makefile emitiendo el mandato configure. Especifique las funciones y extensiones que desea incluir en la versión personalizada de PHP. Un mandato de configuración típico incluye las siguientes opciones:

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

Las opciones de configuración tienen los siguientes efectos:

**--enable-cli**

Habilita la modalidad de línea de mandatos del acceso de PHP.

**--disable-cgi**

Inhabilita la modalidad Common Gateway Interface (CGI) del acceso de PHP.

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

Habilita la modalidad de Apache 2 Dynamic Server Object (DSO) del acceso de PHP.

**--with-zlib**

Habilita el soporte de compresión de zlib.

**--with-pdo-ibm=<sqllib>**

Habilita el controlador de pdo\_ibm utilizando la biblioteca de CLI de DB2 para acceder a los sistemas de bases de datos. El valor <sqllib> hace referencia al directorio en el que se instala DB2.

5. Compile los archivos emitiendo el mandato make.
6. Instale los archivos emitiendo el mandato make install. En función de cómo haya configurado el directorio de instalación de PHP utilizando el mandato configure, es posible que necesite autorización de root para emitir satisfactoriamente este mandato. Esto debería instalar los archivos ejecutables y actualizar la configuración de Apache HTTP Server para que dé soporte a PHP.
7. Instale la extensión ibm\_db2 emitiendo el siguiente mandato como usuario con autorización root:
 

```
pecl install ibm_db2
```

 Este mandato descarga, configura, compila e instala la extensión ibm\_db2 para PHP.
8. Copie el archivo php.ini-recommended en la vía de acceso del archivo de configuración correspondiente a la nueva instalación de PHP. Para determinar la vía de acceso del archivo de configuración, emita el mandato php -i y busque la palabra clave php.ini. Cambie el nombre del archivo por php.ini.

9. Abra el nuevo archivo `php.ini` con un editor de texto y añada las siguientes líneas, donde *instancia* es el nombre de la instancia de DB2 en Linux o UNIX.
  - Para establecer el entorno de DB2 para `pdo_ibm`:  
`PDO_IBM.db2_instance_name=instancia`
  - (Linux o UNIX) Para habilitar la extensión `ibm_db2` y establecer el entorno de DB2:  
`extension=ibm_db2.so`  
`ibm_db2.instance_name=instancia`
10. Vuelva a iniciar Apache HTTP Server para habilitar la configuración modificada.



---

## Capítulo 3. Desarrollo con `ibm_db2`

---

### Conexión a una base de datos DB2 con PHP (`ibm_db2`)

Debe conectarse a una base de datos de DB2 antes de poder crear, actualizar, suprimir o recuperar datos de dicha fuente de datos. La extensión `ibm_db2` para PHP le permite conectarse a una base de datos de DB2 utilizando una conexión catalogada o una conexión TCP/IP directa al sistema de gestión de bases de datos de DB2. También puede crear conexiones persistentes con una base de datos. Las conexiones persistentes mejoran el rendimiento manteniendo la conexión abierta entre peticiones PHP y reutilizando la conexión cuando un script PHP subsiguiente solicita una conexión con un conjunto de credenciales idéntico.

Antes de conectarse a una base de datos de DB2 mediante la extensión `ibm_db2`, debe configurar el entorno de PHP en el sistema y habilitar la extensión `ibm_db2`.

1. Cree una conexión a la base de datos de DB2:

- Para crear una conexión no permanente a una base de datos de DB2, llame a `db2_connect()` con un valor *database* que especifique un nombre de base de datos catalogada o de toda una serie de conexión de base de datos para una conexión TCP/IP directa.
- Para crear una conexión persistente a una base de datos de DB2, llame a `db2_pconnect()` con un valor *database* que especifique un nombre de base de datos catalogada o de toda una serie de conexión de base de datos para una conexión TCP/IP directa.

2. Compruebe el valor devuelto por `db2_connect()` o `db2_pconnect`.

- Si el valor devuelto por `db2_connect()` o `db2_pconnect` es `FALSE`, el intento de conexión ha fallado. Puede recuperar información de diagnóstico mediante `db2_conn_error()` y `db2_conn_errormsg()`.
- Si el valor devuelto por `db2_connect()` o `db2_pconnect` no es `FALSE`, el intento de conexión ha sido satisfactorio. Puede utilizar el recurso de conexión para crear, actualizar, suprimir o recuperar datos con otras funciones de `ibm_db2`.

Cuando crea una conexión llamando a `db2_connect()`, PHP cierra la conexión con la base de datos:

- cuando se llama a `db2_close()` para la conexión,
- cuando se establece el recurso de conexión en `NULL`,
- o cuando finaliza el script de PHP.

Cuando crea una conexión llamando a `db2_pconnect()`, PHP ignora cualquier llamada a `db2_close()` para el recurso de conexión especificado y mantiene la conexión con la base de datos para los siguientes scripts de PHP.

---

### Recuperación de metadatos de la base de datos(`ibm_db2`)

Algunas clases de aplicaciones como, por ejemplo, las interfaces de administración, necesitan reflejar de forma dinámica la estructura y los objetos SQL contenidos en bases de datos arbitrarias. Un enfoque para recuperar metadatos sobre una base de datos consiste en emitir sentencias `SELECT` directamente frente a las tablas de catálogos del sistema; sin embargo, el esquema de las tablas de catálogos del

sistema puede variar entre versiones de DB2 o el esquema de las tablas de catálogos del sistema en DB2 Database para Linux, UNIX y Windows puede diferir del esquema de las tablas de catálogos del sistema en DB2 para z/OS. En vez del laborioso mantenimiento de estas diferencias en el código de aplicación, la extensión `ibm_db2` para PHP ofrece un conjunto estándar de funciones que pueden devolver metadatos a las bases de datos que utilizan DB2 Database para Linux, UNIX y Windows, Cloudscape y DB2 Connect, DB2 para z/OS y DB2 para i5/OS.

- Debe configurar el entorno de PHP del sistema y habilitar la extensión `ibm_db2`.
- Debe tener un recurso de conexión devuelto desde `db2_connect()` o `db2_pconnect()`.

1. Llame a la función que devuelve los metadatos que necesita:

**db2\_client\_info()**

Devuelve los metadatos sobre la configuración y el software del cliente de DB2.

**db2\_column\_privileges()**

Lista las columnas y los privilegios asociados para una tabla.

**db2\_columns()**

Lista las columnas y los metadatos asociados para una tabla.

**db2\_foreign\_keys()**

Lista las claves foráneas para una tabla.

**db2\_primary\_keys()**

Lista las claves primarias para una tabla.

**db2\_procedure\_columns()**

Lista los procedimientos para uno o más procedimientos almacenados.

**db2\_procedures()**

Lista los procedimientos almacenados registrados en la base de datos.

**db2\_server\_info()**

Devuelve los metadatos sobre la configuración y el software del sistema de gestión de bases de datos.

**db2\_special\_columns()**

Lista los identificadores de fila exclusivos para una tabla.

**db2\_statistics()**

Lista los índices y las estadísticas para una tabla.

**db2\_table\_privileges()**

Lista las tablas y sus privilegios asociados en la base de datos.

Tenga en cuenta que aunque la mayoría de las funciones de metadatos `ibm_db2` aceptan un calificador o un parámetro de catálogo, este parámetro sólo debe establecerse en un valor que no sea NULL cuando esté conectado.

2. Dependiendo de la función de metadatos a la que haya llamado,

- Las funciones `db2_client_info()` y `db2_server_info()` devuelven directamente un único objeto con propiedades de sólo lectura. Las propiedades de estos objetos pueden utilizarse para crear una aplicación que tenga un comportamiento distinto en función del sistema de gestión de bases de datos al que se conecte. Por ejemplo, en lugar de codificar un límite del denominador común inferior para todos los sistemas de gestión de bases de datos posibles, una aplicación de administración de bases de datos basada en Web creada en la extensión `ibm_db2` podría utilizar la propiedad `db2_server_info()->MAX_COL_NAME_LEN` para visualizar de forma dinámica los campos de texto para denominar las columnas con longitudes máximas que



se correspondan con la longitud máxima de los nombres de columna en el sistema de gestión de bases de datos al que está conectado.

- Las otras funciones de metadatos devuelven conjuntos de resultados con columnas definidas para cada función. Recupere las filas del conjunto de resultados utilizando las funciones `ibm_db2` habituales para este fin.

Observe que al llamar a funciones de metadatos se consumen una cantidad importante de los recursos del sistema de gestión de bases de datos. Si es posible, plantéese modificar los resultados de las llamadas para su utilización posterior.

---

## Ejecución de expresiones XQuery en PHP (`ibm_db2`)

Después de conectar con una base de datos de DB2, el script de PHP estará preparado para emitir expresiones XQuery. Las funciones `db2_exec()` y `db2_execute()` ejecutan sentencias SQL, mediante las cuales se pueden pasar las expresiones XQuery. Un uso típico de `db2_exec()` es establecer el esquema por omisión para la aplicación en un archivo de inclusión común o clase base.

Debe configurar el entorno de PHP del sistema y habilitar la extensión `ibm_db2`.

Para evitar la amenaza en la seguridad mediante ataques por inyección, `db2_exec()` sólo se debe utilizar para ejecutar sentencias de SQL compuestas de series estáticas. La interpolación de variables PHP que representan la entrada del usuario en la expresión de XQuery puede exponer la aplicación a ataques de inyección.

1. Llame a `db2_exec()` con los argumentos siguientes:
  - a. El recurso de conexión;
  - b. Una serie que contenga la sentencia SQL, incluyendo la expresión XQuery. La expresión XQuery ha de estar envuelta por una cláusula `XMLQUERY` de la sentencia SQL.
  - c. (Opcional): una matriz que contenga opciones de sentencia

### **DB2\_ATTR\_CASE**

Para conseguir compatibilidad con sistemas de bases de datos que no sigan el estándar SQL, esta opción establece el caso en los que los nombres de columna se devolverán a la aplicación. Por omisión, el caso está establecido en `DB2_CASE_NATURAL`, que devuelve nombres de columna tal y como los devuelve DB2. Puede establecer este parámetro en `DB2_CASE_LOWER` para obligar a que los nombres de columna estén en minúsculas, o en `DB2_CASE_UPPER` para obligar a que los nombres de columna estén en mayúsculas.

### **DB2\_ATTR\_CURSOR**

Esta opción establece el tipo de cursor que `ibm_db2` devuelve para los conjuntos de resultados. Por omisión, `ibm_db2` devuelve un cursor de sólo avance (`DB2_FORWARD_ONLY`) que devuelve la fila siguiente del conjunto de resultados para cada llamada a `db2_fetch_array()`, `db2_fetch_assoc()`, `db2_fetch_both()`, `db2_fetch_object()` o `db2_fetch_row()`. Puede establecer este parámetro en `DB2_SCROLLABLE` para solicitar un cursor desplazable de forma que las funciones de captación de `ibm_db2` acepten un segundo argumento especificando la posición absoluta de la fila a la que desea acceder en el conjunto de resultados.

2. Compruebe el valor devuelto por `db2_exec()`:

- Si el valor es FALSE, la sentencia SQL falla. Puede recuperar información de diagnóstico mediante las funciones `db2_stmt_error()` y `db2_stmt_errormsg()`.
- Si el valor no es FALSE, la sentencia de SQL ha sido satisfactoria y ha devuelto un recurso de sentencia que se puede utilizar en llamadas de función subsiguientes relacionadas con esta consulta.

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

---

## Ejecución de sentencias de SQL

### Ejecución de una única sentencia SQL en PHP (ibm\_db2)

Después de conectar con una base de datos de DB2, la mayoría de scripts de PHP ejecutarán uno o más sentencias SQL. La función de `db2_exec()` ejecuta una única sentencia SQL que no acepta parámetros de entrada. Un uso típico de `db2_exec()` es establecer el esquema por omisión para la aplicación en un archivo de inclusión común o clase base.

Debe configurar el entorno de PHP del sistema y habilitar la extensión `ibm_db2`.

Para evitar la amenaza en la seguridad mediante ataques por inyección de SQL, `db2_exec()` sólo se debe utilizar para ejecutar sentencias SQL compuestas de series estáticas. La interpolación de variables de PHP que representan la entrada del usuario en la sentencia SQL puede exponer la aplicación a ataques de inyección de SQL.

1. Llame a `db2_exec()` con los argumentos siguientes:
  - a. El recurso de conexión;
  - b. Una serie que contenga la sentencia SQL;
  - c. (Opcional): una matriz que contenga opciones de sentencia

#### **DB2\_ATTR\_CASE**

Para conseguir compatibilidad con sistemas de bases de datos que no sigan el estándar SQL, esta opción establece el caso en los que los nombres de columna se devolverán a la aplicación. Por omisión, el caso está establecido en `DB2_CASE_NATURAL`, que devuelve nombres de columna tal y como los devuelve DB2. Puede establecer este parámetro en `DB2_CASE_LOWER` para obligar a que los nombres de columna estén en minúsculas, o en `DB2_CASE_UPPER` para obligar a que los nombres de columna estén en mayúsculas.

#### **DB2\_ATTR\_CURSOR**

Esta opción establece el tipo de cursor que `ibm_db2` devuelve para los conjuntos de resultados. Por omisión, `ibm_db2` devuelve un cursor de sólo avance (`DB2_FORWARD_ONLY`) que devuelve la fila siguiente del conjunto de resultados para cada llamada a `db2_fetch_array()`, `db2_fetch_assoc()`, `db2_fetch_both()`, `db2_fetch_object()` o `db2_fetch_row()`. Puede establecer este parámetro en `DB2_SCROLLABLE` para solicitar un cursor desplazable de forma que las funciones de captación de `ibm_db2` acepten un segundo argumento especificando la posición absoluta de la fila a la que desea acceder en el conjunto de resultados.

2. Compruebe el valor devuelto por `db2_exec()`:

- Si el valor es FALSE, la sentencia SQL falla. Puede recuperar información de diagnóstico mediante las funciones `db2_stmt_error()` y `db2_stmt_errormsg()`.
- Si el valor no es FALSE, la sentencia de SQL ha sido satisfactoria y ha devuelto un recurso de sentencia que se puede utilizar en llamadas de función subsiguientes relacionadas con esta consulta.

Si la sentencia SQL ha seleccionado filas utilizando un cursor desplazable, o ha insertado, actualizado o suprimido filas, puede llamar a `db2_num_rows()` para devolver el número de filas que la sentencia ha devuelto o afectado. Si la sentencia SQL ha devuelto un conjunto de resultados, puede empezar a captar filas.

## Preparación y ejecución de sentencias SQL en PHP (ibm\_db2)

La mayoría de sentencias SQL en aplicaciones PHP utilizan entrada de variables para determinar los resultados de la sentencia SQL. Para pasar las entradas proporcionadas por el usuario a una sentencia SQL de forma segura, prepare una sentencia utilizando marcadores de parámetro (?) que representen la entrada de variables. Cuando ejecuta la sentencia preparada, vincula los valores de entrada a los marcadores de entrada. El motor de la base de datos garantiza que cada valor de entrada se trate como un único parámetro, evitando ataques de inyección de SQL contra la aplicación. En comparación con las sentencias emitidas mediante `db2_exec()`, las sentencias preparadas ofrecen una ventaja en el rendimiento porque el sistema de gestión de bases de datos crea un plan de acceso para cada sentencia preparada que puede ser reutilizada si la sentencia se vuelve a emitir posteriormente.

Debe configurar el entorno de PHP del sistema y habilitar la extensión `ibm_db2`.

Sólo puede utilizar marcadores de parámetro como lugar reservado para los valores de columna o predicado. El compilador SQL no podría crear un plan de acceso para una sentencia que usara marcadores de parámetro en lugar de nombres de columna, nombres de tabla u otros identificadores SQL.

Para preparar y ejecutar una sentencia SQL:

1. Llame a `db2_prepare()` con los argumentos siguientes:
  - a. El recurso de conexión
  - b. Una serie que contenga la sentencia SQL, incluidos los marcadores de parámetro (?) para cualquier valor de columna o predicado que requiera entrada de variables
  - c. (Opcional): una matriz que contenga opciones de sentencia

### **DB2\_ATTR\_CASE**

Para conseguir compatibilidad con sistemas de bases de datos que no sigan el estándar SQL, esta opción establece el caso en los que los nombres de columna se devolverán a la aplicación. Por omisión, el caso está establecido en `DB2_CASE_NATURAL`, que devuelve nombres de columna tal y como los devuelve DB2. Puede establecer este parámetro en `DB2_CASE_LOWER` para obligar a que los nombres de columna estén en minúsculas, o en `DB2_CASE_UPPER` para obligar a que los nombres de columna estén en mayúsculas.

### **DB2\_ATTR\_CURSOR**

Esta opción establece el tipo de cursor que `ibm_db2` devuelve para los conjuntos de resultados. Por omisión, `ibm_db2` devuelve un cursor de sólo avance (`DB2_FORWARD_ONLY`) que devuelve la fila

siguiente del conjunto de resultados para cada llamada a `db2_fetch_array()`, `db2_fetch_assoc()`, `db2_fetch_both()`, `db2_fetch_object()` o `db2_fetch_row()`. Puede establecer este parámetro en `DB2_SCROLLABLE` para solicitar un cursor desplazable de forma que las funciones de captación de `ibm_db2` acepten un segundo argumento especificando la posición absoluta de la fila a la que desea acceder en el conjunto de resultados.

2. Compruebe el valor devuelto por `db2_prepare()`.
  - Si el valor es `FALSE`, la sentencia SQL falla. Puede recuperar información de diagnóstico mediante las funciones `db2_stmt_error()` y `db2_stmt_errormsg()`.
  - Si el valor no es `FALSE`, la sentencia de SQL ha sido satisfactoria y ha devuelto un recurso de sentencia que se puede utilizar en llamadas de función subsiguientes relacionadas con esta consulta.
3. (Opcional): llame a `db2_bind_param()` para cada marcador de parámetro en la sentencia SQL con los argumentos siguientes:
  - a. El recurso de sentencia
  - b. Un entero que representa la posición del marcador de parámetro en la sentencia SQL
  - c. Valor que se debe utilizar en lugar del marcador de parámetro
4. Llame a `db2_execute` con los argumentos siguientes:
  - a. El recurso de sentencia
  - b. (Opcional): una matriz que contiene los valores que se deben utilizar en lugar de los marcadores de parámetro, en orden

```
$sql = "SELECT firstnme, lastname FROM employee WHERE bonus > ? AND bonus < ?";
$stmt = db2_prepare($conn, $sql);
if (!$stmt) {
    // Manejar los errores
}

// Vincular explícitamente los parámetros
db2_bind_param($stmt, 1, $_POST['lower']);
db2_bind_param($stmt, 2, $_POST['upper']);

db2_execute($stmt);
// Procesar los resultados

// Invocar de nuevo sentencia preparada usando parámetros vinculados dinámicamente
db2_execute($stmt, array($_POST['lower'], $_POST['upper']));
```

Si ejecuta una sentencia preparada que devuelve uno o más conjuntos de resultados, puede empezar a recuperar filas del recurso de sentencia llamando a las funciones `db2_fetch_array()`, `db2_fetch_assoc()`, `db2_fetch_both()`, `db2_fetch_object()` o `db2_fetch_row()`.

## Inserción de objetos grandes en PHP (`ibm_db2`)

La extensión `ibm_db2` proporciona soporte a la totalidad de la gama de tipos de datos de DB2, incluidos los datos de tipo objeto grande de tipo carácter (CLOB) y objeto grande binario (BLOB). Cuando se inserta un objeto grande en una base de datos, es posible tratar el objeto grande simplemente como una serie PHP. Sin embargo, el tratamiento de un objeto grande como una serie PHP es un enfoque que consume más recursos del servidor PHP de los necesarios. En lugar de cargar todos los datos del objeto grande en una serie PHP y entonces pasarlos a DB2 mediante una sentencia `INSERT`, puede insertar los objetos grandes directamente desde un archivo del servidor PHP.

Debe configurar el entorno de PHP del sistema y habilitar la extensión `ibm_db2`.

Para insertar un objeto grande en la base de datos directamente desde un archivo:

1. Llame a `db2_prepare()` para preparar una sentencia `INSERT` con un marcador de parámetro que represente la columna del objeto grande.
2. Establezca el valor de una variable de PHP en la vía de acceso y el nombre del archivo que contenga los datos del objeto grande. La vía de acceso puede ser relativa o absoluta y está sujeta a los permisos de acceso del ejecutable de PHP.
3. Llame a `db2_bind_param()` para vincular el marcador de parámetro al archivo que contenga los datos del objeto grande. El tercer parámetro es una serie que representa el nombre de la variable de PHP que incluye el nombre del archivo que contiene los datos del objeto grande. El cuarto parámetro es `DB2_PARAM_FILE`, que indica a la extensión `ibm_db2` que recupere los datos de un archivo.
4. Llame a `db2_execute()` para emitir la sentencia `INSERT` y vincular en la base de datos los datos del archivo.

```
$stmt = db2_prepare($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);
```

---

## Lectura de conjuntos de resultados de consultas

### Captación de columnas del conjunto de resultados en PHP (ibm\_db2)

Cuando ejecuta una sentencia que devuelve uno o más conjuntos de resultados, normalmente necesita iterar por las filas devueltas de cada conjunto de resultados. Si el conjunto de resultados incluye columnas con datos extremadamente grande (como, por ejemplo, columnas definidas con un tipo de datos `BLOB` o `CLOB`), es posible que prefiera recuperar los datos en de columna en columna para evitar que el proceso PHP utilice demasiada memoria.

- Debe configurar el entorno de PHP del sistema y habilitar la extensión `ibm_db2`.
  - Debe tener un recurso de sentencia que se haya devuelto desde `db2_exec()` o `db2_execute()` con uno o más conjuntos de resultados asociados.
1. Llame a la función `db2_fetch_row()` para hacer que el cursor avance hasta la fila siguiente del conjunto de resultados. La primera vez que llame a una función de captación para un conjunto de resultados determinado, el cursor se moverá a la primera fila del conjunto de resultados. Si ha solicitado un cursor desplazable, también puede especificar el número de la fila del conjunto de resultados que desea recuperar.
  2. Compruebe el resultado devuelto por `db2_fetch_row()`. Si el resultado es `FALSE`, no hay más filas en el conjunto de resultados.
  3. Llame a la función `db2_result()` para recuperar el valor de la columna solicitada pasando un entero que represente la posición de la columna en la fila (empezando por 0 para la primera columna) o una serie que represente el nombre de la columna.

```
<?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";
}
?>

```

## Captación de filas del conjunto de resultados en PHP (ibm\_db2)

Cuando ejecuta una sentencia que devuelve uno o más conjuntos de resultados, normalmente necesita iterar por las filas devueltas.

- Debe configurar el entorno de PHP del sistema y habilitar la extensión `ibm_db2`.
- Debe tener un recurso de sentencia que se haya devuelto desde `db2_exec()` o `db2_execute()` con uno o más conjuntos de resultados asociados.

Llame a la función de captación de `ibm_db2` que devuelve los datos de la fila en el formato que prefiera:

### **db2\_fetch\_array()**

Devuelve una matriz que contiene los datos correspondientes a las columnas de la fila indexada por posición de columna empezando por 0

### **db2\_fetch\_assoc()**

Devuelve una matriz que contiene los datos correspondientes a las columnas de la fila indexada por nombre de columna.

### **db2\_fetch\_both()**

Devuelve una matriz que contiene los datos correspondientes a las columnas de la fila indexada tanto por nombre de columna como por posición de columna empezando por 0.

### **db2\_fetch\_object()**

Devuelve un objeto que contiene los datos de la fila. El objeto contiene propiedades que coinciden con los nombres de columna de la fila que, cuando se accede a la misma, devuelve los valores correspondientes de las columnas.

Debe pasar el recurso de sentencia como el primer argumento. Si ha solicitado un cursor desplazable al ejecutar `db2_exec()` o `db2_prepare()`, puede pasar un número de fila absoluto como segundo argumento. Con el cursor de sólo avance por omisión, cada llamada a un método de captación devuelve la fila siguiente del conjunto de resultados. Puede continuar captando filas hasta que el método de captación devuelva `FALSE`, lo que significa que ha llegado al final del conjunto de resultados.

```

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

```

## Captación de objetos grandes en PHP (ibm\_db2)

La extensión `ibm_db2` proporciona soporte a la totalidad de la gama de tipos de datos de DB2, incluidos los datos de tipo objeto grande de tipo carácter (CLOB) y objeto grande binario (BLOB). Cuando se capta un objeto grande desde un conjunto de resultados, es posible tratar el objeto grande simplemente como una serie PHP. Sin embargo, el tratamiento de un objeto grande como una serie PHP es un enfoque que consume más recursos del servidor PHP de los necesarios. Si el

objetivo final es crear un archivo que contenga los datos para un objeto grande, puede ahorrar recursos del sistema captando los objetos grandes directamente en un archivo del servidor PHP.

Debe configurar el entorno de PHP del sistema y habilitar la extensión `ibm_db2`.

Para captar un objeto grande desde la base de datos directamente en un archivo:

1. Cree una variable PHP que represente una corriente. Por ejemplo, el valor de retorno de una llamada a `fopen()`.
2. Llame a `db2_prepare()` para crear una sentencia `SELECT`.
3. Llame a `db2_bind_param()` para vincular la columna de salida para el objeto grande con la variable de PHP que representa la corriente. El tercer parámetro es una serie que representa el nombre de la variable de PHP que incluye el nombre del archivo que contiene los datos del objeto grande. El cuarto parámetro es `DB2_PARAM_FILE`, que indica a la extensión `ibm_db2` que grabe los datos en un archivo.
4. Llame a `db2_execute()` para emitir la sentencia SQL.
5. Llame a una función de captación `ibm_db2` de su elección (por ejemplo, `db2_fetch_object()`), para recuperar la fila siguiente del conjunto de resultados.

```
$stmt = db2_prepare($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);
```

---

## Gestión de transacciones en PHP (`ibm_db2`)

Por omisión, la extensión `ibm_db2` abre cada conexión en modalidad `Autocommit`. La modalidad `Autocommit` le ayuda a evitar problemas de escalas de bloqueo que puedan obstaculizar el rendimiento de aplicaciones Web muy escalables. Sin embargo, en algunos scripts es posible que necesite retrotraer una transacción que contenga una o varias sentencias SQL. La extensión `ibm_db2` le permite ejercer un control detallado de las transacciones.

Debe configurar el entorno de PHP del sistema y habilitar la extensión `ibm_db2`.

Debe utilizar una conexión regular creada con `db2_connect()` para controlar las transacciones de base de datos en PHP. Las conexiones persistentes siempre utilizan la modalidad `Autocommit`.

Para iniciar una transacción:

1. Cree una conexión de base de datos utilizando el valor `"AUTOCOMMIT" => DB2_AUTOCOMMIT_OFF` de la matriz de opciones `db2_connect()`. También puede desactivar `Autocommit` para un recurso de conexión existente llamando a `db2_autocommit($conn, DB2_AUTOCOMMIT_OFF)`. Llamar a `db2_autocommit()` requiere comunicación adicional de PHP al sistema de gestión de bases de datos y puede afectar al rendimiento de los scripts PHP.
2. Emita una o más sentencias SQL en el ámbito de la transacción de base de datos utilizando el recurso de conexión para el que se han habilitado las transacciones.
3. Confirme o retrotraiga la transacción:
  - Para confirmar la transacción, llame a `db2_commit()`.

- Para retrotraer la transacción, llame a `db2_rollback()`.
4. (Opcional): vuelva a poner la conexión de base de datos en modalidad Autocommit llamando a `db2_autocommit($conn, DB2_AUTOCOMMIT_ON)`. Si emite otra sentencia SQL sin volver a poner la conexión de base de datos en modalidad Autocommit, empieza una nueva transacción que requerirá una confirmación o retrotracción.

Si emite sentencias SQL en una transacción y el script finaliza sin confirmar o retrotraer explícitamente la transacción, la extensión `ibm_db2` retrotrae automáticamente cualquier trabajo realizado en la transacción.

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

// Emitir una o más sentencias SQL en la transacción
$result = db2_exec($conn, 'DELETE FROM TABLE employee');
if ($result === FALSE) {
    print '<p>No se ha podido completar la transacción</p>';
    db2_rollback($conn);
}
else {
    print '<p>Successfully completed transaction!</p>';
    db2_commit($conn);
}
```

---

## Gestión de errores y mensajes de aviso (`ibm_db2`)

En ocasiones un usuario experimenta problemas al intentar conectarse a una base de datos o emitir una sentencia SQL. Es posible que la contraseña para la conexión sea incorrecta, que la tabla a la que hace referencia en una sentencia `SELECT` no exista o que la sintaxis para una sentencia SQL no sea válida. Deberá efectuar una codificación defensiva y utilizar las funciones de gestión de errores que ofrece la extensión `ibm_db2` para permitir a la aplicación recuperarse de forma satisfactoria del problema.

Debe configurar el entorno de PHP del sistema y habilitar la extensión `ibm_db2`.

1. Compruebe el valor devuelto desde la función `ibm_db2` para asegurarse de que el valor devuelto por la función sea satisfactorio. Si la función puede devolver el valor 0 como, por ejemplo, `db2_num_rows()`, debe probar de forma explícita si el valor era `FALSE` utilizando el operador de PHP `===`.
2. Si la función ha devuelto `FALSE` en lugar del valor del recurso de conexión, el recurso de sentencia o el valor numérico que esperaba, llame a la función de gestión de errores de `ibm_db2` que se adecue al contexto de la aplicación y a las necesidades de la aplicación:

### Errores de conexión

Para recuperar el `SQLSTATE` devuelto por el último intento de conexión, llame a `db2_conn_error()`. Para recuperar un mensaje de error descriptivo adecuado al registro de errores de la aplicación, llame a `db2_conn_errormsg()`.

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

### Errores de SQL (ejecución de sentencias SQL directamente y captación de resultados)

Para recuperar el `SQLSTATE` devuelto por el último intento de preparar



o ejecutar una sentencia SQL o para captar un resultado de un conjunto de resultados, llame a `db2_stmt_error()`. Para recuperar un mensaje de error descriptivo adecuado al registro de errores de la aplicación, llame a `db2_stmt_errormsg()`.

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

#### **Errores de SQL (ejecución de sentencias preparadas)**

Si el valor devuelto por `db2_prepare()` ha sido satisfactorio pero una llamada posterior a `db2_execute()` no es satisfactoria, llame a `db2_stmt_error()` o `db2_stmt_errormsg()` y pase a `db2_prepare()` el recurso devuelto por la llamada como el argumento.

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

3. Para evitar que se produzcan posibles vulneraciones de la seguridad como resultado de la visualización directa del `SQLSTATE` devuelto por la base de datos y ofrecer una mejor experiencia general del usuario en la aplicación Web, utilice una estructura de conmutación para recuperar de los estados de errores conocidos o devolver mensajes de errores personalizados.

```
switch($this->state):
    case '22001':
        // Más datos de los permitidos para la columna definida
        $message = "Ha entrado demasiados caracteres para este valor.";
        break;
```

---

## **Invocación de procedimientos almacenados**

### **Llamada a procedimientos almacenados con parámetros OUT o INOUT en PHP (ibm\_db2)**

DB2 proporciona soporte a los procedimientos almacenados con parámetros que sólo acepten un valor de entrada (parámetros IN), que sólo devuelvan un valor de salida (parámetros OUT) o que acepten un valor de entrada y devuelvan un valor de salida (INOUT). Con la extensión `ibm_db2` para PHP, los parámetros IN pueden gestionarse como cualquier otro marcador de parámetro en una sentencia SQL. Sin embargo, la extensión `ibm_db2` también permite LLAMAR a los procedimientos almacenados con parámetros OUT e INOUT y recuperar los valores de salida de dichos parámetros.

Debe configurar el entorno de PHP del sistema y habilitar la extensión `ibm_db2`.

Para llamar a un procedimiento almacenado con parámetros OUT o INOUT:

1. Llame a `db2_prepare()` para preparar una sentencia `CALL` con marcadores de parámetros que representen los parámetros OUT e INOUT.
2. Llame a `db2_bind_param()` para vincular cada marcador de parámetro con el nombre de la variable de PHP que contendrá el valor de salida del parámetro después de emitir la sentencia `CALL`. Para los parámetros INOUT, el valor de la variable de PHP se pasa como el valor de entrada del parámetro cuando se

emite la sentencia CALL. Establezca el cuarto parámetro de `db2_bind_param()` en `DB2_PARAM_OUT`, que representa un parámetro OUT, o en `DB2_PARAM_INOUT`, que representa un parámetro INOUT.

3. Llame a `db2_execute()` para emitir la sentencia CALL y vincular en las variables de PHP los datos del procedimiento almacenado.

```
$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";
```

## Llamada a procedimientos almacenados que devuelven varios conjuntos de resultados en PHP (ibm\_db2)

DB2 permite crear y llamar procedimientos almacenados que devuelven más de un conjunto de resultados. La extensión `ibm_db2` para PHP proporciona soporte completo a esta posibilidad mediante la función `db2_next_result()`. Puede utilizar esta función para captar filas de distintos conjuntos de resultados que una sola llamada haya devuelto al mismo procedimiento almacenado en el orden que prefiera.

- Debe configurar el entorno de PHP del sistema y habilitar la extensión `ibm_db2`.
- Debe tener un recurso de sentencia que se haya devuelto desde la llamada a un procedimiento almacenado con `db2_exec()` o `db2_execute()`.

Para devolver varios conjuntos de resultados desde un procedimiento almacenado:

1. El primer conjunto de resultados está asociado con el recurso de sentencia que la sentencia CALL ha devuelto.
2. Pase el recurso de sentencia original a `db2_next_result()` como el primer argumento para recuperar el segundo conjunto de resultados y los conjuntos de resultados siguientes. Esta función devuelve FALSE cuando no hay disponibles más conjuntos de resultados.

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

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

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

print "\nFetching third result set\n";
$result_3 = db2_next_result($stmt);
```

```
if ($result_3) {  
    while ($row = db2_fetch_array($result_3)) {  
        // Trabajar con fila  
    }  
}
```



---

## Capítulo 4. Desarrollo con PDO

---

### Conexión a una base de datos DB2 con PHP (PDO)

Debe conectarse a una base de datos de DB2 antes de poder crear, actualizar, suprimir o recuperar datos de dicha fuente de datos. La interfaz de Objetos de datos PHP (PDO) para PHP le permite conectarse a una base de datos de DB2 utilizando una conexión catalogada o una conexión TCP/IP directa al sistema de gestión de bases de datos de DB2 mediante la extensión PDO\_IBM. Además, puede crear conexiones persistentes a una fuente de datos que mejoran el rendimiento manteniendo la conexión abierta entre peticiones PHP y reutilizando la conexión cuando un script PHP subsiguiente solicita una conexión con un conjunto de credenciales idéntico.

Debe configurar el entorno de PHP 5.1 o superior en el sistema y habilitar las extensiones PDO y PDO\_IBM.

1. Cree una conexión a la base de datos de DB2 llamando al constructor PDO en un bloque `try{}`. Pase un valor *DSN* que especifique `ibm:` para la extensión PDO\_IBM, seguido de un nombre de base de datos catalogada o de toda una serie de conexión de base de datos para una conexión TCP/IP directa.
  - (Windows): Por omisión, PDO\_IBM utiliza el agrupamiento de conexiones para minimizar los recursos de conexión y mejorar el rendimiento de las conexiones. .
  - (Linux y UNIX): PDO\_IBM ofrece conexiones permanentes si pasa `array(PDO::ATTR_PERSISTENT => TRUE)` como el cuarto argumento del constructor PDO.
2. (Opcional): establezca las opciones de manejo de errores para la conexión PDO en el cuarto argumento al constructor PDO:
  - Por omisión, PDO establece un mensaje de error que se puede recuperar mediante `PDO::errorInfo()` y un `SQLCODE` que se puede recuperar mediante `PDO::errorCode()` cuando se produce cualquier error; para solicitar esta modalidad de forma explícita, establezca `PDO::ATTR_ERRMODE => PDO::ERRMODE_SILENT`
  - Para emitir un `E_WARNING` de PHP cuando se produzca cualquier error, además de establecer el mensaje de error y `SQLCODE`, establezca `PDO::ATTR_ERRMODE => PDO::ERRMODE_WARNING`
  - Para emitir una excepción de PHP cuando se produzca cualquier error, establezca `PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION`
3. Detecte cualquier excepción emitida por el bloque `try{}` en un bloque `catch {}` correspondiente.

```
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());
}
```

Cuando crea una conexión mediante PDO, PHP cierra la conexión a la base de datos:

- cuando establece el objeto de PDO en NULL,
- o cuando finaliza el script de PHP.

---

## Ejecución de sentencias de SQL

### Ejecución de una única sentencia SQL en PHP que no devuelve ningún conjunto de resultados (PDO)

Después de conectar con una base de datos de DB2, la mayoría de scripts de PHP ejecutarán uno o más sentencias SQL. El método `PDO::exec()` ejecuta una única sentencia SQL que no acepta ningún parámetro de entrada y no devuelve ningún conjunto de resultados. Un uso típico de `PDO::exec()` es establecer el esquema por omisión para la aplicación en un archivo de inclusión común o clase base.

Debe configurar el entorno de PHP del sistema y habilitar la extensión `PDO_IBM`.

Para evitar la amenaza en la seguridad mediante ataques por inyección SQL, `PDO::exec()` sólo se debe utilizar para ejecutar sentencias SQL compuestas de series estáticas. La interpolación de variables de PHP que representan la entrada del usuario en la sentencia SQL puede exponer la aplicación a ataques de inyección de SQL.

Para ejecutar una única sentencia SQL en PHP:

1. Llame al método `PDO::exec()` en el objeto de conexión PDO con una serie que contenga la sentencia SQL.
2. Si la sentencia SQL ha insertado, modificado o suprimido filas, `PDO::exec()` devuelve un valor entero que representa el número de filas que se han insertado, modificado o suprimido. Para determinar si `PDO::exec()` ha devuelto `FALSE` indicando una condición de error o `0` indicando que no se ha insertado, modificado o suprimido ninguna fila, debe utilizar el operador `===` para probar estrictamente el valor devuelto contra `FALSE`.

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

### Ejecución de una única sentencia SQL en PHP que devuelve un conjunto de resultados (PDO)

Después de conectar con una base de datos de DB2, la mayoría de scripts de PHP ejecutarán uno o más sentencias SQL. El método `PDO::query()` ejecuta una única sentencia SQL que no acepta ningún parámetro de entrada y devuelve uno o más conjuntos de resultados. El uso típico de `PDO::query()` es ejecutar una sentencia `SELECT` estática.

Debe configurar el entorno de PHP del sistema y habilitar la extensión `PDO_IBM`.

Para evitar la amenaza en la seguridad mediante ataques por inyección de SQL, sólo se debe utilizar `PDO::query()` para ejecutar sentencias SQL compuestas de series estáticas. La interpolación de variables de PHP que representan la entrada del usuario en la sentencia SQL puede exponer la aplicación a ataques de inyección de SQL.

Para ejecutar una única sentencia SQL en PHP que devuelva un conjunto de resultados:

1. Llame al método `PDO::query()` en el objeto de conexión PDO con una serie que contenga la sentencia SQL.
2. Compruebe el valor devuelto por `PDO::query()`.
  - Si el valor es `FALSE`, la sentencia SQL falla. Puede recuperar información de diagnóstico mediante los métodos `PDO::errorCode()` y `PDO::errorInfo()`.
  - Si el valor no es `FALSE`, la sentencia SQL ha sido satisfactoria y ha devuelto un recurso `PDOStatement` que se puede utilizar en llamadas de método subsiguientes.

```
$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>";
}
```

Después de crear un objeto `PDOStatement` con `PDO::query()`, puede empezar inmediatamente a recuperar filas del objeto con los métodos `PDOStatement::fetch()` o `PDOStatement::fetchAll()`.

## Preparación y ejecución de sentencias SQL (PDO)

La mayoría de sentencias SQL en aplicaciones PHP utilizan entrada de variables para determinar los resultados de la sentencia SQL. Para pasar las entradas proporcionadas por el usuario a una sentencia SQL de forma segura, prepare una sentencia utilizando marcadores de parámetro (?) o variables con nombre que representen la entrada de variables. Cuando ejecuta la sentencia preparada, vincula los valores de entrada a los marcadores de entrada. El motor de la base de datos garantiza que cada valor de entrada se trate como un único parámetro, evitando ataques de inyección de SQL contra la aplicación. En comparación con las sentencias emitidas mediante `PDO::exec()`, las sentencias preparadas ofrecen una ventaja en el rendimiento porque el sistema de gestión de bases de datos crea un plan de acceso para cada sentencia preparada que puede ser reutilizada si la sentencia se vuelve a emitir posteriormente.

Debe configurar el entorno de PHP del sistema y habilitar la extensión `PDO_IBM`.

- Sólo puede utilizar marcadores de parámetro como lugar reservado para los valores de columna o predicado. El compilador SQL no podría crear un plan de acceso para una sentencia que usara marcadores de parámetro en lugar de nombres de columna, nombres de tabla u otros identificadores SQL.
- No puede utilizar en la misma sentencia SQL marcadores de parámetro (?) y marcadores de parámetro con nombre (:name).

Para preparar y ejecutar una sentencia SQL:

1. Llame a `PDO::prepare()` con los argumentos siguientes:
  - a. Una serie que contenga la sentencia SQL, incluidos los marcadores de parámetro (?) o variables con nombre (:name) para cualquier valor de columna o predicado que requiera entrada de variables
  - b. (Opcional): una matriz que contenga opciones de sentencia

### **PDO::ATTR\_CURSOR**

Esta opción establece el tipo de cursor que PDO devuelve para los

conjuntos de resultados. Por omisión, PDO devuelve un cursor de sólo avance (*PDO::CURSOR\_FWDONLY*) que devuelve la fila siguiente de un conjunto de resultados para cada llamada a *PDOStatement::fetch()*. Puede establecer este parámetro en *PDO::CURSOR\_SCROLL* para solicitar un cursor desplazable.

2. Compruebe el valor devuelto por *PDO::prepare()*.
  - Si el valor es *FALSE*, la sentencia SQL falla. Puede recuperar información de diagnóstico mediante los métodos *PDO::errorCode()* y *PDO::errorInfo()*.
  - Si el valor no es *FALSE*, la sentencia SQL ha sido satisfactoria y ha devuelto un objeto *PDOStatement* que se puede utilizar en llamadas de método subsiguientes.
3. (Opcional): llame a *PDOStatement::bindParam()* para cada marcador de parámetro en la sentencia SQL con los argumentos siguientes:
  - a. Identificador de parámetro. Para marcadores de parámetro de signo de interrogación (?), éste es un entero que representa la posición indexada 1 del parámetro en la sentencia SQL. Para marcadores de parámetro con nombre (:name), ésta es una serie que representa el nombre de parámetro.
  - b. Valor que se debe utilizar en lugar del marcador de parámetro
4. Llame a *PDOStatement::execute()*, opcionalmente pasando una matriz que contiene los valores que se deben utilizar en lugar de los marcadores de parámetro, en orden para marcadores de parámetro de signo de interrogación o como una matriz asociativa *:name => value* para marcadores de parámetro con nombre.

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

```
// Vincular explícitamente los parámetros
$stmt->bindParam(1, $_POST['lower']);
$stmt->bindParam(2, $_POST['upper']);
```

```
$stmt->execute($stmt);
```

```
// Invocar de nuevo la sentencia utilizando parámetros vinculados dinámicamente
$stmt->execute($stmt, array($_POST['lower'], $_POST['upper']));
```

Puede ejecutar de forma satisfactoria una sentencia preparada que devuelve uno o más conjuntos de resultados, puede empezar a recuperar filas del recurso de sentencia llamando a los métodos *PDOStatement::fetch()* o *PDOStatement::fetchAll()*.

## Inserción de objetos grandes en PHP (PDO)

La extensión PDO proporciona soporte a la totalidad de la gama de tipos de datos de DB2, incluidos los datos de tipo objeto grande de tipo carácter (CLOB) y objeto grande binario (BLOB). Cuando se inserta un objeto grande en una base de datos, es posible tratar el objeto grande simplemente como una serie PHP. Sin embargo, el tratamiento de un objeto grande como una serie PHP es un enfoque que consume más recursos del servidor PHP de los necesarios. En lugar de cargar todos los datos del objeto grande en una serie PHP y entonces pasarlos a DB2 mediante una sentencia *INSERT*, puede insertar los objetos grandes directamente desde un archivo del servidor PHP.



Debe configurar el entorno de PHP 5.1 o superior en el sistema y habilitar las extensiones PDO y PDO\_IBM.

Para insertar un objeto grande en la base de datos directamente desde un archivo:

1. Llame a `PDO::prepare()` para crear un objeto `PDOStatement` a partir de una sentencia `INSERT` con un marcador de parámetro que represente la columna del objeto grande.
2. Cree una variable de PHP que represente una corriente—por ejemplo, el valor de retorno de una llamada a `fopen()`.
3. Llame a `PDOStatement::bindParam()` para vincular el marcador de parámetro a la variable de PHP que represente la corriente de datos para el objeto grande. El tercer parámetro es una serie que representa el nombre de la variable de PHP que incluye el nombre del archivo que contiene los datos del objeto grande. El cuarto parámetro es una constante de PHP, `PDO::PARAM_LOB`, que indica a la extensión PDO que debe recuperar los datos de un archivo.
4. Llame a `PDOStatement::execute()` para emitir la sentencia `INSERT` y vincular en la base de datos los datos del archivo.

```
$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();
```

---

## Lectura de conjuntos de resultados de consultas

### Captación de columnas de conjuntos de resultados en PHP (PDO)

Cuando ejecuta una sentencia que devuelve uno o más conjuntos de resultados, normalmente necesita iterar por las filas devueltas de cada conjunto de resultados. En algunos casos, sólo necesita devolver una sola columna de cada fila en el conjunto de resultados. Aunque podría reescribir una sentencia `SELECT` para este fin, podría carecer de los privilegios necesarios para reescribir un procedimiento almacenado que devolviera más columnas que necesita.

- Debe configurar el entorno de PHP en el sistema y habilitar las extensiones PDO y PDO\_IBM.
- Debe tener un recurso de sentencia que se haya devuelto de `PDO::query()` o `PDOStatement::execute()` con uno o más conjuntos de resultados asociados.

Si decide captar una columna desde una fila, en lugar de recuperar todas las columnas de la totalidad de la fila de forma simultánea, puede devolver una sola columna de cada fila.

- Para devolver una sola columna desde una sola fila del conjunto de resultados:  
Llame al método `PDOStatement::fetchColumn()` especificando la columna que desee recuperar como el primer argumento del método. Los números de columna empiezan por el 0. Si no especifica ninguna columna, `PDOStatement::fetchColumn()` devuelve la primera columna de la fila.
- Para devolver una matriz que contenga una sola columna de todas las filas restantes del conjunto de resultados:

Llame al método `PDOStatement::fetchAll()` pasando `PDO::FETCH_COLUMN` como el primer argumento y la columna que desea recuperar como el segundo argumento, para devolver una matriz de los valores para la columna seleccionada del conjunto de resultados. Los números de columna empiezan por

el 0. Si no especifica ninguna columna, `PDOStatement::fetchAll(PDO::FETCH_COLUMN)` devuelve la primera columna de la fila.

## Captación de filas de conjuntos de resultados en PHP (PDO)

Cuando ejecuta una sentencia que devuelve uno o más conjuntos de resultados, normalmente necesita iterar por las filas devueltas.

- Debe configurar el entorno de PHP del sistema y habilitar la extensión PDO.
- Debe tener un objeto `PDOStatement` que se haya devuelto desde `PDO::query()` o `PDOStatement::execute()` con uno o más conjuntos de resultados asociados.

Para devolver una única fila de un conjunto de resultados como una matriz u objeto, llame al método `PDOStatement::fetch()`.

Para devolver todas las filas del conjunto de resultados como una matriz de matrices u objetos, llame al método `PDOStatement::fetchAll()`.

Por omisión, PDO devuelve cada fila como una matriz indexada por nombre de columna y una posición de columna indexada 0 en la fila. Puede solicitar un estilo de devolución diferente pasando una de las constantes siguientes como primer parámetro de `PDOStatement::fetch()`:

### **PDO::FETCH\_ASSOC**

Devuelve una matriz indexada por nombre de columna tal y como se ha devuelto en el conjunto de resultados.

### **PDO::FETCH\_BOTH (valor por omisión)**

Devuelve una matriz indexada tanto por nombre de columna como por número de columna indexado 0 en el conjunto de resultados

### **PDO::FETCH\_BOUND**

Devuelve TRUE y asigna los valores de las columnas del conjunto de resultados a las variables de PHP a las que estaban vinculadas con el método `PDOStatement::bindParam()`.

### **PDO::FETCH\_CLASS**

Devuelve una nueva instancia de la clase solicitada, correlacionando las columnas del conjunto de resultados con las propiedades con nombre de la clase.

### **PDO::FETCH\_INTO**

Actualiza una instancia existente de la clase solicitada, correlacionando las columnas del conjunto de resultados con las propiedades de la clase.

### **PDO::FETCH\_LAZY**

Combina `PDO::FETCH_BOTH` y `PDO::FETCH_OBJ`, creando los nombres de variable de objeto conforme se accede a los mismos.

### **PDO::FETCH\_NUM**

Devuelve una matriz indexada por número de columna tal y como se ha devuelto en el conjunto de resultados, empezando en la columna 0.

### **PDO::FETCH\_OBJ**

Devuelve un objeto anónimo con nombres de propiedad que se corresponden con los nombres de columna devueltos en el conjunto de resultados.

(Opcional): si ha solicitado un cursor desplazable cuando ha llamado a `PDO::query()` o `PDOStatement::execute()`, puede pasar dos argumentos adicionales a `PDOStatement::fetch()`:

1. La orientación de captación para esta petición de captación:

**PDO::FETCH\_ORI\_NEXT (valor por omisión)**

Capta la siguiente fila del conjunto de resultados.

**PDO::FETCH\_ORI\_PRIOR**

Capta la fila anterior del conjunto de resultados.

**PDO::FETCH\_ORI\_FIRST**

Capta la primera fila del conjunto de resultados.

**PDO::FETCH\_ORI\_LAST**

Capta la última fila del conjunto de resultados.

**PDO::FETCH\_ORI\_ABS**

Capta la fila absoluta del conjunto de resultados. Requiere un entero positivo como tercer argumento en `PDOStatement::fetch()`.

**PDO::FETCH\_ORI\_REL**

capta la fila relativa del conjunto de resultados. Requiere un entero positivo o negativo como tercer argumento de `PDOStatement::fetch()`.

2. Entero que solicita la fila absoluta o relativa del conjunto de resultados, correspondiente a la orientación de captación solicitada en el segundo argumento de `PDOStatement::fetch()`.

`PDOStatement::fetch()` devuelve `FALSE` cuando se ha recuperado la última fila del conjunto de resultados para un conjunto de resultados de sólo avance.

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

## Captación de objetos grandes en PHP (PDO)

La extensión PDO proporciona soporte a la totalidad de la gama de tipos de datos de DB2, incluidos los datos de tipo objeto grande de tipo carácter (CLOB) y objeto grande binario (BLOB). Cuando se capta un objeto grande desde un conjunto de resultados, es posible tratar el objeto grande simplemente como una serie PHP. Sin embargo, el tratamiento de un objeto grande como una serie PHP es un enfoque que consume más recursos del servidor PHP de los necesarios. Si el objetivo final es crear un archivo que contenga los datos para un objeto grande, puede ahorrar recursos del sistema captando los objetos grandes directamente en un archivo del servidor PHP.

Debe configurar el entorno de PHP 5.1 o superior en el sistema y habilitar las extensiones PDO y PDO\_IBM.

Para captar un objeto grande desde la base de datos directamente en un archivo:

1. Cree una variable de PHP que represente una corriente—por ejemplo, el valor de retorno de una llamada a `fopen()`.
2. Llame a `PDO::prepare()` para crear un objeto `PDOStatement` desde una sentencia SQL.
3. Llame a `PDOStatement::bindColumn()` para vincular la columna de salida para el objeto grande con la variable de PHP que representa la corriente. El tercer parámetro es una serie que representa el nombre de la variable de PHP que

incluye el nombre del archivo que contiene los datos del objeto grande. El cuarto parámetro es una constante PHP, `PDO::PARAM_LOB`, que indica a la extensión PDO que escriba datos en un archivo. Observe que debe llamar a `PDOStatement::bindColumn()` para asignar una variable de PHP distinta para cada columna del conjunto de resultados.

4. Llame a `PDOStatement::execute()` para emitir la sentencia SQL.
5. Llame a `PDOStatement::fetch(PDO::FETCH_BOUND)` para recuperar la fila siguiente del conjunto de resultados, vinculando la salida de la columna en las variables de PHP que haya asociado con el método `PDOStatement::bindColumn()`.

```
$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);
```

---

## Gestión de transacciones en PHP (PDO)

Por omisión, PDO abre cada conexión en modalidad Autocommit. La modalidad Autocommit le ayuda a evitar problemas de escalas de bloqueo que puedan obstaculizar el rendimiento de aplicaciones Web muy escalables. Sin embargo, en algunos scripts es posible que necesite retrotraer una transacción que contenga una o varias sentencias SQL. PDO le permite ejercer un control detallado de las transacciones.

Debe configurar el entorno de PHP del sistema y habilitar la extensión PDO.

Para iniciar una transacción:

1. Llame a `PDO::beginTransaction()` para iniciar una nueva transacción.
2. Emita una o más sentencias SQL en el ámbito de la transacción de base de datos utilizando el recurso de conexión para el que se han habilitado las transacciones.
3. Confirme o retrotraiga la transacción:
  - Para confirmar la transacción, llame a `PDO::commit()`.
  - Para retrotraer la transacción, llame a `PDO::rollback()`.

Después de confirmar o retrotraer la transacción, PDO restablece automáticamente la conexión de base de datos a la modalidad Autocommit. Si emite sentencias SQL en una transacción y el script finaliza sin confirmar o retrotraer explícitamente la transacción, PDO automáticamente retrotrae cualquier trabajo realizado en la transacción.

```
$conn = new PDO('ibm:SAMPLE', 'db2inst1', 'ibmdb2', array(
    PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION));
// PDO::ERRMODE_EXCEPTION significa que un error SQL emite una excepción
try {
    // Emitir estas sentencias SQL en una transacción en un bloque try{}
    $conn->beginTransaction();

    // Una o más sentencias SQL

    $conn->commit();
}
catch (Exception $e) {
    // Si algo ha emitido una excepción en nuestro bloque de transacciones
```

```
// de sentencias, retrotraer cualquier trabajo realizado en la transacción
print '<p>No se ha podido completar la transacción</p>';
$conn->rollBack();
}
```

---

## Manejo de errores y avisos en PHP (PDO)

En ocasiones un usuario experimenta problemas al intentar conectarse a una base de datos o emitir una sentencia SQL. Es posible que la contraseña para la conexión sea incorrecta, que la tabla a la que hace referencia en una sentencia SELECT no exista o que la sintaxis para una sentencia SQL no sea válida. Deberá efectuar una codificación defensiva y utilizar las funciones de manejo de errores que ofrece PDO para permitir a la aplicación recuperarse de forma satisfactoria del problema.

Debe configurar el entorno de PHP en el sistema y habilitar las extensiones PDO y PDO\_IBM.

PDO le ofrece la opción de manejar errores como avisos, errores o excepciones. Sin embargo, cuando crea un nuevo objeto de conexión de PDO, este siempre emite un objeto `PDOException` si se produce un error. Si no capta la excepción, PHP imprime un rastreo de la información del error, que puede mostrar las credenciales de conexión de la base de datos, incluido su nombre de usuario y contraseña.

- Para captar un objeto `PDOException` y manejar el error asociado:
  1. Transmite la llamada al constructor de PDO en un bloque `try`.
  2. A continuación del bloque `try`, incluya un bloque `catch` que capte el objeto `PDOException`.
  3. Recupere el mensaje de error asociado al error invocando el método `Exception::getMessage()` en el objeto `PDOException`.
- Para recuperar el SQLSTATE asociado a un objeto PDO o `PDOStatement`, invoque el método `errorCode()` sobre el objeto.
- Para recuperar una matriz de información del error asociada a un objeto PDO o `PDOStatement`, invoque el método `errorInfo()` sobre el objeto. La matriz contiene una serie que representa el SQLSTATE como el primer elemento, un entero que representa el código de error SQL o CLI como el segundo elemento y una serie que contiene el texto completo del mensaje de error como el tercer elemento.

---

## Invocación de procedimientos almacenados

### Llamada a procedimientos almacenados con parámetros OUT o INOUT en PHP (PDO)

DB2 proporciona soporte a los procedimientos almacenados con parámetros que sólo acepten un valor de entrada (parámetros IN), que sólo devuelvan un valor de salida (parámetros OUT) o que acepten un valor de entrada y devuelvan un valor de salida (INOUT). Con la extensión PDO\_IBM para PHP, los parámetros IN pueden gestionarse como cualquier otro marcador de parámetro en una sentencia SQL. Sin embargo, la extensión PDO\_IBM también permite LLAMAR a los procedimientos almacenados con parámetros OUT e INOUT y recuperar los valores de salida de dichos parámetros.

Debe configurar el entorno de PHP en el sistema y habilitar las extensiones PDO y PDO\_IBM.

Para llamar a un procedimiento almacenado con parámetros OUT o INOUT:

1. Llame a `PDO::prepare()` para preparar una sentencia CALL con marcadores de parámetros que representen los parámetros OUT e INOUT.
2. Llame a `PDOStatement::bindParam()` para vincular cada marcador de parámetro con el nombre de la variable de PHP que contendrá el valor de salida del parámetro después de emitir la sentencia CALL. Para los parámetros INOUT, el valor de la variable de PHP se pasa como el valor de entrada del parámetro cuando se emite la sentencia CALL. Establezca el tercer parámetro de `PDOStatement::bindParam()` en el tipo de datos que se está vinculando:

**PDO::PARAM\_NULL**

Representa el tipo de datos SQL NULL.

**PDO::PARAM\_INT**

Representa los tipos enteros SQL.

**PDO::PARAM\_LOB**

Representa los tipos de objetos grandes SQL.

**PDO::PARAM\_STR**

Representa los tipos de datos de caracteres SQL.

3. Para un parámetro INOUT, utilice el operador basado en bits OR para añadir `PDO::PARAM_INPUT_OUTPUT` al tipo de datos que se está vinculando.
4. Establezca el cuarto parámetro de `PDOStatement::bindParam()` en la longitud máxima esperada del valor de salida.

```
$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";
```

## Llamada a procedimientos almacenados que devuelven varios conjuntos de resultados en PHP (PDO)

DB2 permite crear y llamar procedimientos almacenados que devuelven más de un conjunto de resultados. La extensión PDO\_IBM para PHP proporciona soporte a esta posibilidad mediante el método `nextRowset()`. Puede utilizar este método para captar filas de distintos conjuntos de resultados que una sola llamada haya devuelto al mismo procedimiento almacenado.

- Debe configurar el entorno de PHP 5.1 o superior en el sistema y habilitar la extensión PDO.
- Debe tener un objeto `PDOStatement` que se haya devuelto desde la llamada a un procedimiento almacenado con `PDO::query()` o `PDOStatement::execute()`.

Para devolver varios conjuntos de resultados desde un procedimiento almacenado:

1. El primer conjunto de resultados está asociado con el objeto PDOStatement que la sentencia CALL ha devuelto. Puede captar filas desde el objeto PDOStatement hasta que no haya disponibles más filas en el primer conjunto de resultados.
2. Llame al método nextRowset() del objeto PDOStatement para devolver el siguiente conjunto de resultados. Puede captar filas desde el objeto PDOStatement hasta que no haya disponibles más filas en el siguiente conjunto de resultados.

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





---

## Parte 2. Desarrollo de aplicaciones Perl



---

## Capítulo 5. Consideraciones sobre la programación en Perl

Perl es un lenguaje de programación popular que está disponible libremente para muchos sistemas operativos. Si utiliza el controlador DBD::DB2 disponible en <http://www.ibm.com/software/data/db2/perl> con el módulo DBI de interfaz de base de datos de Perl que puede encontrar en <http://www.perl.com>, puede crear aplicaciones DB2 utilizando Perl.

Dado que Perl es un lenguaje interpretado y que el Módulo DBI de Perl utiliza SQL dinámico, Perl constituye el lenguaje ideal para crear y revisar con rapidez los prototipos de aplicaciones DB2. El Módulo DBI de Perl utiliza una interfaz que es muy parecida a las interfaces CLI y JDBC, lo cual facilita el transporte de los prototipos de Perl a CLI y JDBC.

La mayoría de proveedores de bases de datos proporcionan un controlador de base de datos para el Módulo DBI de Perl, lo cual significa que también se puede utilizar Perl para crear aplicaciones que accedan a datos de muchos servidores de bases de datos distintos. Por ejemplo, puede escribir en Perl una aplicación DB2 que conecte con una base de datos Oracle utilizando el controlador de base de datos DBD::Oracle, captar datos de la base de datos Oracle e insertar los datos en una base de datos DB2 utilizando el controlador de base de datos DBD::DB2.

---

### Conexiones de bases de datos en Perl

Para permitir que Perl cargue el módulo DBI, debe incluir la línea siguiente en la aplicación DB2:

```
use DBI;
```

El módulo DBI carga automáticamente el controlador DBD::DB2 cuando se crea un *descriptor de contexto de base de datos* utilizando la sentencia DBI->connect con la sintaxis siguiente:

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

donde:

**\$dbhhandle**

representa el descriptor de contexto de base de datos devuelto por la sentencia de conexión

**dbalias**

representa un alias de DB2 catalogado en el directorio de bases de datos de DB2

**\$userID**

representa el ID de usuario utilizado para conectar con la base de datos

**\$password**

representa la contraseña para el ID de usuario utilizado para conectar con la base de datos

---

## Captación de resultados en Perl

Puesto que el Módulo DBI de Perl sólo soporta SQL dinámico, no puede utilizar variables del lenguaje principal en las aplicaciones DB2 en Perl.

Para devolver resultados de una consulta de SQL, lleve a cabo los pasos siguientes:

1. Cree un descriptor de contexto de base de datos estableciendo conexión con la base de datos con la sentencia `DBI->connect`.
2. Cree un descriptor de contexto de sentencia a partir del descriptor de contexto de base de datos. Por ejemplo, puede llamar a `prepare` con una sentencia de SQL como argumento de serie para devolver el descriptor de contexto de sentencia `$sth`, tal como se muestra en la sentencia de Perl siguiente:

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

3. Ejecute la sentencia de SQL llamando a `execute` en el descriptor de contexto de sentencia. Una llamada satisfactoria a `execute` asocia un conjunto de resultados al descriptor de contexto de sentencia. Por ejemplo, puede ejecutar la sentencia preparada en el ejemplo anterior utilizando la sentencia de Perl siguiente:

```
#Nota: $rc representa el código de retorno de la llamada a execute
my $rc = $sth->execute;
```

4. Capte una fila del conjunto de resultados asociado al descriptor de contexto de sentencia mediante una llamada a `fetchrow()`. El DBI de Perl devuelve una fila en forma de matriz con un valor por columna. Por ejemplo, puede devolver todas las filas del descriptor de contexto de sentencia del ejemplo anterior utilizando la sentencia de Perl siguiente:

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

---

## Marcadores de parámetros en Perl

Para permitirle ejecutar una sentencia preparada utilizando distintos valores de entrada para campos específicos, el módulo DBI de Perl le permite preparar y ejecutar una sentencia utilizando marcadores de parámetros. Para incluir un marcador de parámetros en una sentencia de SQL, utilice un signo de interrogación (?)

El código Perl siguiente crea un descriptor de contexto de sentencia que acepta un marcador de parámetros para la cláusula `WHERE` de una sentencia `SELECT`. A continuación, el código ejecuta la sentencia dos veces, utilizando los valores de entrada 25000 y 35000 para sustituir el marcador de parámetros.

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

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

.
.
.

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

---

## Variables SQLSTATE y SQLCODE en Perl

Para devolver el SQLSTATE asociado a un descriptor de contexto de base de datos del DBI de Perl o a un descriptor de contexto de sentencia, llame al método `state`. Por ejemplo, para devolver el SQLSTATE asociado al descriptor de contexto de base de datos `$descripbd`, incluya la sentencia de Perl siguiente en la aplicación:

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

Para devolver el SQLCODE asociado a un descriptor de contexto de base de datos del DBI de Perl o a un descriptor de contexto de sentencia, llame al método `err`. Para devolver el mensaje para un SQLCODE asociado a un descriptor de contexto de base de datos del DBI de Perl o a un descriptor de contexto de sentencia, llame al método `errstr`. Por ejemplo, para devolver el SQLCODE asociado al descriptor de contexto de base de datos `$descripbd`, incluya la sentencia de Perl siguiente en la aplicación:

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

---

## Restricciones de Perl

El módulo DBI de Perl sólo soporta SQL dinámico. Cuando sea necesario ejecutar una sentencia varias veces, se puede mejorar el rendimiento de las aplicaciones DB2 en Perl emitiendo una llamada `prepare` para preparar la sentencia.

Perl no da soporte al acceso a bases de datos de varias hebras.

Para obtener información actual sobre las restricciones de la versión del controlador de `DBD::DB2` que instale en la estación de trabajo, consulte el archivo `CAVEATS` contenido en el paquete del controlador de `DBD::DB2`.

---

## Ejemplo de programa Perl

A continuación se muestra un ejemplo de una aplicación escrita en Perl:

```
#!/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 firstnme, 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 (($firstnme, $lastname) = $sth->fetchrow()) {
    print "$firstnme: $lastname\n";
}

# comprobar si hay problemas que puedan haber cancelado antes la captación
```

```
warn $DBI::errstr if $DBI::err;

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

---

## Creación de aplicaciones Perl

DB2 da soporte al acceso a bases de datos para aplicaciones cliente escritas en Perl 5.8. En el momento de imprimir el presente manual, el release 1.0 del controlador de DB2 (DBD::DB2) correspondiente a Perl Database Interface (Perl DBI) versión 1.41 o posterior recibe soporte y está disponible para AIX, HP-UX, Linux, Solaris y Windows. Para tener información sobre cómo obtener el controlador más reciente, visite <http://www.ibm.com/db2/perl>.

DB2 proporciona programas de ejemplo en Perl, ubicados en el directorio `sqllib/samples/perl` para UNIX y en el directorio `sqllib\samples\perl` para Windows.

Para ejecutar el intérprete de Perl sobre un programa Perl desde la línea de mandatos, entre el nombre del intérprete y el nombre del programa (incluyendo la extensión):

1. Si está conectando localmente en el servidor:

```
perl dbauth.pl
```

2. Si está conectando desde un cliente remoto:

```
perl dbauth.pl sample <IDusuario> <contraseña>
```

Algunos programas requieren archivos de soporte para que se puedan ejecutar. El programa de ejemplo `tbssel` requiere varias tablas creadas por el script `tbsselcreate.db2` del CLP. El script `tbsselinit` (UNIX), o el archivo de proceso por lotes `tbsselinit.bat` (Windows), en primer lugar llama a `tbsseldrop.db2` para eliminar las tablas, en caso de que existan, y luego llama a `tbsselcreate.db2` para crearlas. Por lo tanto, para ejecutar el programa deberá entrar los mandatos siguientes:

1. Si está conectando localmente en el servidor:

```
tbsselinit
perl tbssel.pl
```

2. Si está conectando desde un cliente remoto:

```
tbsselinit
perl tbssel.pl sample <IDusuario> <contraseña>
```

**Nota:** Para un cliente remoto es necesario modificar la sentencia de conexión en los archivos `tbsselinit` o `tbsselinit.bat` para codificar el ID de usuario y la contraseña: `db2 connect to sample user <IDusuario> using <contraseña>`

---

## Parte 3. Apéndices





---

## Apéndice A. Visión general de la información técnica de DB2

La información técnica de DB2 está disponible a través de las herramientas y los métodos siguientes:

- Centro de información de DB2
  - Temas (Tareas, concepto y temas de consulta)
  - Ayuda para herramientas de DB2
  - Programas de ejemplo
  - Guías de aprendizaje
- Manuales de DB2
  - Archivos PDF (descargables)
  - Archivos PDF (desde el DVD con PDF de DB2)
  - Manuales en copia impresa
- Ayuda de línea de mandatos
  - Ayuda de mandatos
  - Ayuda de mensajes

**Nota:** Los temas del Centro de información de DB2 se actualizan con más frecuencia que los manuales en PDF o impresos. Para obtener la información más actualizada, instale las actualizaciones de la documentación cuando estén disponibles, o consulte el Centro de información de DB2 en [ibm.com](http://ibm.com).

Puede acceder a información técnica adicional de DB2 como, por ejemplo, notas técnicas, documentos técnicos y publicaciones IBM Redbooks en línea, en el sitio [ibm.com](http://ibm.com). Acceda al sitio de la biblioteca de software de gestión de información de DB2 en <http://www.ibm.com/software/data/sw-library/>.

### Comentarios sobre la documentación

Agradecemos los comentarios sobre la documentación de DB2. Si tiene sugerencias sobre cómo podemos mejorar la documentación de DB2, envíe un correo electrónico a [db2docs@ca.ibm.com](mailto:db2docs@ca.ibm.com). El personal encargado de la documentación de DB2 lee todos los comentarios de los usuarios, pero no puede responderlos directamente. Proporcione ejemplos específicos siempre que sea posible de manera que podamos comprender mejor sus problemas. Si realiza comentarios sobre un tema o archivo de ayuda determinado, incluya el título del tema y el URL.

No utilice esta dirección de correo electrónico para contactar con el Soporte al cliente de DB2. Si tiene un problema técnico de DB2 que no está tratado por la documentación, consulte al centro local de servicio técnico de IBM para obtener ayuda.

---

## Biblioteca técnica de DB2 en copia impresa o en formato PDF

Las tablas siguientes describen la biblioteca de DB2 que está disponible en el Centro de publicaciones de IBM en [www.ibm.com/shop/publications/order](http://www.ibm.com/shop/publications/order). Los manuales de DB2 Versión 9.5 en inglés en formato PDF y las versiones traducidas se pueden descargar del sitio [www.ibm.com/support/docview.wss?rs=71&uid=swg2700947](http://www.ibm.com/support/docview.wss?rs=71&uid=swg2700947).

Aunque las tablas identifican los manuales en copia impresa disponibles, puede que dichos manuales no estén disponibles en su país o región.

El número de documento se incrementa cada vez que se actualiza un manual. Asegúrese de que lee la versión más reciente de los manuales, tal como aparece a continuación:

**Nota:** El Centro de información de DB2 se actualiza con más frecuencia que los manuales en PDF o impresos.

*Tabla 1. Información técnica de DB2*

<b>Nombre</b>	<b>Número de documento</b>	<b>Copia impresa disponible</b>
<i>Consulta de las API administrativas</i>	SC11-3505-01	Sí
<i>Rutinas y vistas administrativas</i>	SC11-3507-01	No
<i>Call Level Interface Guide and Reference, Volume 1</i>	SC23-5844-01	Sí
<i>Call Level Interface Guide and Reference, Volume 2</i>	SC23-5845-01	Sí
<i>Consulta de mandatos</i>	SC11-3506-01	Sí
<i>Data Movement Utilities Guide and Reference</i>	SC23-5847-01	Sí
<i>Data Recovery and High Availability Guide and Reference</i>	SC23-5848-01	Sí
<i>Data Servers, Databases, and Database Objects Guide</i>	SC23-5849-01	Sí
<i>Database Security Guide</i>	SC23-5850-01	Sí
<i>Desarrollo de aplicaciones ADO.NET y OLE DB</i>	SC11-3499-01	Sí
<i>Desarrollo de aplicaciones de SQL incorporado</i>	SC11-3500-01	Sí
<i>Desarrollo de aplicaciones Java</i>	SC11-3501-01	Sí
<i>Desarrollo de aplicaciones Perl y PHP</i>	SC11-3502-01	No
<i>Desarrollo de rutinas definidas por el usuario (SQL y externas)</i>	SC11-3503-01	Sí
<i>Iniciación al desarrollo de aplicaciones de bases de datos</i>	GC11-3504-01	Sí
<i>Iniciación a la instalación y administración de DB2 en Linux y Windows</i>	GC11-3511-01	Sí
<i>Internationalization Guide</i>	SC23-5858-01	Sí
<i>Consulta de mensajes, Volumen 1</i>	GI11-7823-00	No
<i>Consulta de mensajes, Volumen 2</i>	GI11-7824-00	No
<i>Guía de migración</i>	GC11-3510-01	Sí
<i>Net Search Extender Guía de administración y del usuario</i>	SC11-3615-01	Sí
<i>Partitioning and Clustering Guide</i>	SC23-5860-01	Sí

**Tabla 1. Información técnica de DB2 (continuación)**

<b>Nombre</b>	<b>Número de documento</b>	<b>Copia impresa disponible</b>
<i>Query Patroller Administration and User's Guide</i>	SC23-8507-00	Sí
<i>Guía rápida de iniciación para clientes IBM Data Server</i>	GC11-3513-01	No
<i>Guía rápida de iniciación para servidores DB2</i>	GC11-3512-01	Sí
<i>Spatial Extender and Geodetic Data Management Feature Guía del usuario y manual de consulta</i>	SC11-3614-01	Sí
<i>Consulta de SQL, Volumen 1</i>	SC11-3508-01	Sí
<i>Consulta de SQL, Volumen 2</i>	SC11-3509-01	Sí
<i>System Monitor Guide and Reference</i>	SC23-5865-01	Sí
<i>Troubleshooting Guide</i>	GI11-7857-01	No
<i>Tuning Database Performance</i>	SC23-5867-01	Sí
<i>Guía de aprendizaje de Visual Explain</i>	SC11-3518-00	No
<i>Novedades</i>	SC11-3517-01	Sí
<i>Workload Manager Guide and Reference</i>	SC23-5870-01	Sí
<i>pureXML Guide</i>	SC23-5871-01	Sí
<i>XQuery Reference</i>	SC23-5872-01	No

**Tabla 2. Información técnica específica de DB2 Connect**

<b>Nombre</b>	<b>Número de documento</b>	<b>Copia impresa disponible</b>
<i>Guía rápida de iniciación para DB2 Connect Personal Edition</i>	GC11-3515-01	Sí
<i>Guía rápida de iniciación para servidores DB2 Connect</i>	GC11-3516-01	Sí
<i>Guía del usuario de DB2 Connect</i>	SC11-3514-01	Sí

**Tabla 3. Información técnica de Information Integration**

<b>Nombre</b>	<b>Número de documento</b>	<b>Copia impresa disponible</b>
<i>Information Integration: Administration Guide for Federated Systems</i>	SC19-1020-01	Sí
<i>Information Integration: ASNCLP Program Reference for Replication and Event Publishing</i>	SC19-1018-02	Sí
<i>Information Integration: Configuration Guide for Federated Data Sources</i>	SC19-1034-01	No
<i>Information Integration: SQL Replication Guide and Reference</i>	SC19-1030-01	Sí

Tabla 3. Información técnica de Information Integration (continuación)

Nombre	Número de documento	Copia impresa disponible
<i>Information Integration: Introduction to Replication and Event Publishing</i>	SC19-1028-01	Sí

## Pedido de manuales de DB2 en copia impresa

Si necesita manuales de DB2 en copia impresa, puede comprarlos en línea en varios países o regiones, pero no en todos. Siempre puede hacer pedidos de manuales de DB2 en copia impresa a través del representante local de IBM. Recuerde que algunas publicaciones en copia software del DVD *Documentación en PDF de DB2* no están disponibles en copia impresa. Por ejemplo, no está disponible la publicación *Consulta de mensajes de DB2* en copia impresa.

Las versiones impresas de muchas de las publicaciones de DB2 disponibles en el DVD de Documentación en PDF de DB2 se pueden solicitar a IBM por una cantidad. Dependiendo desde dónde realice el pedido, podrá solicitar manuales en línea, desde el Centro de publicaciones de IBM. Si la realización de pedidos en línea no está disponible en su país o región, siempre puede hacer pedidos de manuales de DB2 en copia impresa al representante local de IBM. Tenga en cuenta que no todas las publicaciones del DVD de Documentación en PDF de DB2 están disponibles en copia impresa.

**Nota:** La documentación más actualizada y completa de DB2 se conserva en el Centro de información de DB2 en <http://publib.boulder.ibm.com/infocenter/db2luw/v9r5>.

Para hacer pedidos de manuales de DB2 en copia impresa:

- Para averiguar si puede hacer pedidos de manuales de DB2 en copia impresa en línea en su país o región, consulte el Centro de publicaciones de IBM en el sitio <http://www.ibm.com/shop/publications/order>. Debe seleccionar un país, región o idioma para poder acceder a la información sobre pedidos de publicaciones y, a continuación, seguir las instrucciones sobre pedidos para su localidad.
- Para hacer pedidos de manuales de DB2 en copia impresa a través del representante local de IBM:
  1. Localice la información de contacto de su representante local desde uno de los siguientes sitios Web:
    - El directorio de IBM de contactos en todo el mundo en el sitio [www.ibm.com/planetwide](http://www.ibm.com/planetwide)
    - El sitio Web de publicaciones de IBM en el sitio <http://www.ibm.com/shop/publications/order>. Tendrá que seleccionar su país, región o idioma para acceder a la página de presentación de las publicaciones apropiadas para su localidad. Desde esta página, siga el enlace "Acerca de este sitio".
  2. Cuando llame, indique que desea hacer un pedido de una publicación de DB2.
  3. Proporcione al representante los títulos y números de documento de las publicaciones que desee solicitar. Si desea consultar los títulos y los números de documento, consulte el apartado "Biblioteca técnica de DB2 en copia impresa o en formato PDF" en la página 43.

---

## Visualización de la ayuda para estados de SQL desde el procesador de línea de mandatos

DB2 devuelve un valor de SQLSTATE para las condiciones que pueden ser el resultado de una sentencia de SQL. La ayuda de SQLSTATE explica los significados de los estados de SQL y los códigos de las clases de estados de SQL.

Para invocar la ayuda para estados de SQL, abra el procesador de línea de mandatos y entre:

```
? sqlstate o ? código de clase
```

donde *sqlstate* representa un estado de SQL válido de cinco dígitos y *código de clase* representa los dos primeros dígitos del estado de SQL.

Por ejemplo, ? 08003 visualiza la ayuda para el estado de SQL 08003, y ? 08 visualiza la ayuda para el código de clase 08.

---

## Acceso a diferentes versiones del Centro de información de DB2

Para los temas de DB2 Version 9.5, el URL del Centro de información de DB2 es <http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/>

Para los temas de DB2 Version 9, el URL del Centro de información de DB2 es <http://publib.boulder.ibm.com/infocenter/db2luw/v9/>

Para los temas de DB2 Version 8, vaya al URL del Centro de información de la Versión 8 en el sitio: <http://publib.boulder.ibm.com/infocenter/db2luw/v8/>

---

## Visualización de temas en su idioma preferido en el Centro de información de DB2

El Centro de información de DB2 intenta visualizar los temas en el idioma especificado en las preferencias del navegador. Si un tema no se ha traducido al idioma preferido, el Centro de información de DB2 visualiza dicho tema en inglés.

- Para visualizar temas en su idioma preferido en el navegador Internet Explorer:

1. En Internet Explorer, pulse en el botón **Herramientas** —> **Opciones de Internet** —> **Idiomas...** Se abrirá la ventana Preferencias de idioma.
2. Asegúrese de que su idioma preferido esté especificado como la primera entrada de la lista de idiomas.
  - Para añadir un nuevo idioma a la lista, pulse el botón **Agregar...**

**Nota:** La adición de un idioma no garantiza que el sistema tenga los fonts necesarios para visualizar los temas en el idioma preferido.

- Para mover un idioma hacia el principio de la lista, seleccione el idioma y pulse el botón **Subir** hasta que el idioma esté en primer lugar en la lista de idiomas.
  - 3. Borre la antememoria del navegador y, a continuación, renueve la página para visualizar el Centro de información de DB2 en su idioma preferido.
- Para visualizar temas en su idioma preferido en un navegador Firefox o Mozilla:
    1. Seleccione el botón en la sección **Idiomas** del diálogo **Herramientas** —> **Opciones** —> **Avanzado**. Se visualizará el panel Idiomas en la ventana Preferencias.

2. Asegúrese de que su idioma preferido esté especificado como la primera entrada de la lista de idiomas.
  - Para añadir un nuevo idioma a la lista, pulse el botón **Añadir...** a fin de seleccionar un idioma en la ventana Añadir idiomas.
  - Para mover un idioma hacia el principio de la lista, seleccione el idioma y pulse el botón **Subir** hasta que el idioma esté en primer lugar en la lista de idiomas.
3. Borre la antememoria del navegador y, a continuación, renueve la página para visualizar el Centro de información de DB2 en su idioma preferido.

En algunas combinaciones de navegador y sistema operativo, puede que también tenga que cambiar los valores regionales del sistema operativo al entorno local y al idioma de su elección.

---

## Actualización del Centro de información de DB2 instalado en el sistema o en el servidor de intranet

Si ha instalado localmente el Centro de información de DB2, puede obtener las actualizaciones de la documentación de IBM e instalarlas.

Para actualizar el Centro de información de DB2 instalado localmente es preciso que:

1. Detenga el Centro de información de DB2 en el sistema, y reinicie el Centro de información en modalidad autónoma. La ejecución del Centro de información en modalidad autónoma impide que otros usuarios de la red accedan al Centro de información y permite al usuario aplicar las actualizaciones. Los Centros de información no administrativos y no root de DB2 se ejecutan siempre en modalidad autónoma.
2. Utilice la función Actualizar para ver qué actualizaciones están disponibles. Si hay actualizaciones que desee instalar, puede utilizar la función Actualizar para obtenerlas e instalarlas

**Nota:** Si su entorno requiere la instalación de actualizaciones del Centro de información de DB2 en una máquina no conectada a Internet, debe duplicar el sitio de actualizaciones en un sistema de archivos local utilizando una máquina que esté conectada a Internet y tenga instalado el Centro de información de DB2. Si muchos usuarios en la red van a instalar las actualizaciones de la documentación, puede reducir el tiempo necesario para realizar las actualizaciones duplicando también el sitio de actualizaciones localmente y creando un proxy para el sitio de actualizaciones.

Si hay paquetes de actualización disponibles, utilice la característica Actualizar para obtener los paquetes. Sin embargo, la característica Actualizar sólo está disponible en modalidad autónoma.

3. Detenga el Centro de información autónomo y reinicie el Centro de información de DB2 en su equipo.

**Nota:** En Windows Vista, los mandatos listados más abajo se deben ejecutar como administrador. Para iniciar un indicador de mandatos o una herramienta gráfica con privilegios de administrador completos, pulse con el botón derecho del ratón el atajo y, a continuación, seleccione **Ejecutar como administrador**.

Para actualizar el Centro de información de DB2 instalado en el sistema o en el servidor de Intranet:

1. Detenga el Centro de información de DB2.

- En Windows, pulse **Inicio** → **Panel de control** → **Herramientas administrativas** → **Servicios**. A continuación, pulse con el botón derecho del ratón en el servicio **Centro de información de DB2** y seleccione **Detener**.
  - En Linux, especifique el mandato siguiente:  
`/etc/init.d/db2icdv95 stop`
2. Inicie el Centro de información en modalidad autónoma.
    - En Windows:
      - a. Abra una ventana de mandatos.
      - b. Navegue hasta la vía de acceso en la que está instalado el Centro de información. De forma predeterminada, el Centro de información de DB2 se instala en el directorio <Archivos de programa>\IBM\Centro de información de DB2\Versión 9.5, siendo <Archivos de programa> la ubicación del directorio Archivos de programa.
      - c. Navegue desde el directorio de instalación al directorio doc\bin.
      - d. Ejecute el archivo help\_start.bat:  
`help_start.bat`
    - En Linux:
      - a. Navegue hasta la vía de acceso en la que está instalado el Centro de información. De forma predeterminada, el Centro de información de DB2 se instala en el directorio /opt/ibm/db2ic/V9.5.
      - b. Navegue desde el directorio de instalación al directorio doc/bin.
      - c. Ejecute el script help\_start:  
`help_start`

Se inicia el navegador Web por omisión del sistema para visualizar el Centro de información autónomo.

3. Pulse en el botón **Actualizar** (🔄). En la derecha del panel del Centro de información, pulse en **Buscar actualizaciones**. Se visualiza una lista de actualizaciones para la documentación existente.
4. Para iniciar el proceso de instalación, compruebe las selecciones que desee instalar y, a continuación, pulse **Instalar actualizaciones**.
5. Cuando finalice el proceso de instalación, pulse **Finalizar**.
6. Detenga el Centro de información autónomo:
  - En Windows, navegue hasta el directorio doc\bin del directorio de instalación y ejecute el archivo help\_end.bat:  
`help_end.bat`

**Nota:** El archivo help\_end de proceso por lotes contiene los mandatos necesarios para concluir sin peligro los procesos que se iniciaron mediante el archivo help\_start de proceso por lotes. No utilice Control-C ni ningún otro método para concluir help\_start.bat.

  - En Linux, navegue hasta el directorio de instalación doc/bin y ejecute el script help\_end:  
`help_end`

**Nota:** El script help\_end contiene los mandatos necesarios para concluir sin peligro los procesos que se iniciaron mediante el script help\_start. No utilice ningún otro método para concluir el script help\_start.
7. Reinicie el Centro de información de DB2:

- En Windows, pulse **Inicio** → **Panel de control** → **Herramientas administrativas** → **Servicios**. A continuación, pulse con el botón derecho del ratón en el servicio **Centro de información de DB2** y seleccione **Iniciar**.
- En Linux, especifique el mandato siguiente:  
`/etc/init.d/db2icdv95 start`

El Centro de información de DB2 actualizado visualiza los temas nuevos y actualizados.

---

## Guías de aprendizaje de DB2

Las guías de aprendizaje de DB2 le ayudan a conocer diversos aspectos de productos DB2. Se proporcionan instrucciones paso a paso a través de lecciones.

### Antes de comenzar

Puede ver la versión XHTML de la guía de aprendizaje desde el Centro de información en el sitio <http://publib.boulder.ibm.com/infocenter/db2help/>.

Algunas lecciones utilizan datos o código de ejemplo. Consulte la guía de aprendizaje para obtener una descripción de los prerrequisitos para las tareas específicas.

### Guías de aprendizaje de DB2

Para ver la guía de aprendizaje, pulse el título.

#### **“pureXML” en *pureXML Guide***

Configure una base de datos DB2 para almacenar datos XML y realizar operaciones básicas con el almacén de datos XML nativos.

#### **“Visual Explain” en *Guía de aprendizaje de Visual Explain***

Analizar, optimizar y ajustar sentencias de SQL para obtener un mejor rendimiento al utilizar Visual Explain.

---

## Información de resolución de problemas de DB2

Existe una gran variedad de información para la resolución y determinación de problemas para ayudarle en la utilización de productos DB2.

### Documentación de DB2

Puede encontrar información sobre la resolución de problemas en la publicación DB2 Troubleshooting Guide o en la sección Soporte y resolución de problemas del Centro de información de DB2. En ellas encontrará información sobre cómo aislar e identificar problemas utilizando herramientas y programas de utilidad de diagnóstico de DB2, soluciones a algunos de los problemas más habituales y otros consejos sobre cómo solucionar problemas que podría encontrar en los productos DB2.

### Sitio web de soporte técnico de DB2

Consulte el sitio Web de soporte técnico de DB2 si tiene problemas y desea obtener ayuda para encontrar las causas y soluciones posibles. El sitio de soporte técnico tiene enlaces a las publicaciones más recientes de DB2, notas técnicas, Informes autorizados de análisis del programa (APAR o arreglos de defectos), fixpacks y otros recursos. Puede buscar en esta base de conocimiento para encontrar posibles soluciones a los problemas.



Acceda al sitio Web de soporte técnico de DB2 en el sitio  
<http://www.ibm.com/software/data/db2/udb/support.html>

---

## Términos y condiciones

Los permisos para utilizar estas publicaciones se otorgan sujetos a los siguientes términos y condiciones.

**Uso personal:** Puede reproducir estas publicaciones para su uso personal, no comercial, siempre y cuando se mantengan los avisos sobre la propiedad. No puede distribuir, visualizar o realizar trabajos derivados de estas publicaciones, o de partes de las mismas, sin el consentimiento expreso de IBM.

**Uso comercial:** Puede reproducir, distribuir y visualizar estas publicaciones únicamente dentro de su empresa, siempre y cuando se mantengan todos los avisos sobre la propiedad. No puede realizar trabajos derivados de estas publicaciones, ni reproducirlas, distribuirlas o visualizarlas, ni de partes de las mismas fuera de su empresa, sin el consentimiento expreso de IBM.

Excepto lo expresamente concedido en este permiso, no se conceden otros permisos, licencias ni derechos, explícitos o implícitos, sobre las publicaciones ni sobre ninguna información, datos, software u otra propiedad intelectual contenida en el mismo.

IBM se reserva el derecho de retirar los permisos aquí concedidos cuando, a su discreción, el uso de las publicaciones sea en detrimento de su interés o cuando, según determine IBM, las instrucciones anteriores no se cumplan correctamente.

No puede descargar, exportar ni volver a exportar esta información excepto en el caso de cumplimiento total con todas las leyes y regulaciones vigentes, incluyendo todas las leyes y regulaciones sobre exportación de los Estados Unidos.

IBM NO GARANTIZA EL CONTENIDO DE ESTAS PUBLICACIONES. LAS PUBLICACIONES SE PROPORCIONAN "TAL CUAL" Y SIN GARANTÍA DE NINGUNA CLASE, NI EXPLÍCITA NI IMPLÍCITA, INCLUYENDO PERO SIN LIMITARSE A LAS GARANTÍAS IMPLÍCITAS DE COMERCIALIZACIÓN, NO VULNERACIÓN E IDONEIDAD PARA UN FIN DETERMINADO.



---

## Apéndice B. Avisos

Esta información ha sido desarrollada para productos y servicios que se ofrecen en Estados Unidos de América

Es posible que IBM no comercialice en otros países algunos productos, servicios o características descritos en este manual. Consulte al representante local de IBM para obtener información sobre los productos y servicios que actualmente pueden adquirirse en su zona. Cualquier referencia a un producto, programa o servicio de IBM no pretende afirmar ni implicar que sólo se pueda utilizar dicho producto, programa o servicio de IBM. En su lugar se puede utilizar cualquier producto, programa o servicio funcionalmente equivalente que no vulnere ninguno de los derechos de propiedad intelectual de IBM. Sin embargo, es responsabilidad del usuario evaluar y verificar el funcionamiento de cualquier producto, programa o servicio que no sea de IBM.

IBM puede tener patentes o solicitudes de patentes en tramitación que afecten al tema tratado en este documento. La posesión de este documento no confiere ninguna licencia sobre dichas patentes. Puede realizar consultas sobre licencias escribiendo a:

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
EE.UU.

Para realizar consultas sobre licencias referentes a información de doble byte (DBCS), puede ponerse en contacto con el Departamento de Propiedad Intelectual de IBM de su país/región o escribir a:

IBM World Trade Asia Corporation  
Licensing  
2-31 Roppongi 3-chome, Minato-ku  
Tokio 106, Japón

**El párrafo siguiente no es aplicable al Reino Unido ni a ningún país/región en donde tales disposiciones sean incompatibles con la legislación local:**

INTERNATIONAL BUSINESS MACHINES CORPORATION PROPORCIONA ESTA PUBLICACIÓN "TAL CUAL", SIN GARANTÍA DE NINGUNA CLASE, NI EXPLÍCITA NI IMPLÍCITA, INCLUIDAS, PERO SIN LIMITARSE A ELLAS, LAS GARANTÍAS IMPLÍCITAS DE NO VULNERACIÓN DE DERECHOS, COMERCIALIZACIÓN O IDONEIDAD PARA UN FIN DETERMINADO. Algunos estados no permiten la exclusión de garantías expresas o implícitas en determinadas transacciones, por lo que es posible que esta declaración no sea aplicable en su caso.

Esta publicación puede contener inexactitudes técnicas o errores tipográficos. Periódicamente se efectúan cambios en la información aquí contenida; dichos cambios se incorporarán a las nuevas ediciones de la publicación. IBM puede efectuar, en cualquier momento y sin previo aviso, mejoras y cambios en los productos y programas descritos en esta publicación.

Este documento puede proporcionar enlaces o referencias a sitios y recursos que no son de IBM. IBM no representa, no da garantías, ni se compromete con los recursos de terceros ni con los recursos que no son de IBM a los cuales se puede hacer referencia, acceder desde o enlazarse con desde este documento. Un enlace a un sitio que no es de IBM no implica que IBM apruebe el contenido o la utilización de dicho sitio Web o a su propietario. Además, IBM no forma parte ni es responsable de ninguna transacción que el usuario pueda realizar con terceros, aún cuando llegue a conocerlos (o utilice un enlace a ellos) desde un sitio de IBM. De acuerdo a esto, el usuario reconoce y acepta que IBM no es responsable de la disponibilidad de dichos recursos o sitios externos ni tampoco es responsable de ningún contenido, servicio, producto u otros materiales que estén o se encuentren disponibles desde dichos sitios o recursos. Cualquier software que proporcionen terceras partes, estarán sujetos a los términos y condiciones de licencia que acompañen al software.

IBM puede utilizar o distribuir cualquier información que se le facilite de la manera que considere adecuada, sin contraer por ello ninguna obligación con el remitente.

Los licenciarios de este programa que deseen obtener información sobre él con el fin de habilitar: (i) el intercambio de información entre programas creados de forma independiente y otros programas (incluido éste) y (ii) el uso mutuo de la información intercambiada, deben ponerse en contacto con:

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

Dicha información puede estar disponible, sujeta a los términos y condiciones apropiados, incluido en algunos casos el pago de una tarifa.

El programa bajo licencia descrito en este documento y todo el material bajo licencia asociado a él, los proporciona IBM según los términos del Acuerdo de Cliente de IBM, el Acuerdo Internacional de Programas Bajo Licencia de IBM o cualquier acuerdo equivalente entre el usuario e IBM.

Los datos de rendimiento contenidos en este documento se obtuvieron en un entorno controlado. Por lo tanto, los resultados obtenidos en otros entornos operativos pueden variar significativamente. Algunas mediciones pueden haberse realizado en sistemas experimentales y no es seguro que estas mediciones sean las mismas en los sistemas disponibles comercialmente. Además, algunas mediciones pueden haberse calculado mediante extrapolación. Los resultados reales pueden variar. Los usuarios del presente manual deben verificar los datos aplicables para su entorno específico.

La información referente a productos que no son de IBM se ha obtenido de los proveedores de esos productos, de sus anuncios publicados o de otras fuentes disponibles públicamente. IBM no ha probado esos productos y no puede confirmar la exactitud del rendimiento, la compatibilidad ni ninguna otra afirmación referente a productos que no son de IBM. Las preguntas sobre las prestaciones de productos que no son de IBM deben dirigirse a los proveedores de esos productos.

Todas las declaraciones de intenciones de IBM están sujetas a cambio o cancelación sin previo aviso, y sólo representan objetivos.

Este manual puede contener ejemplos de datos e informes que se utilizan en operaciones comerciales diarias. Para ilustrarlos de la forma más completa posible, los ejemplos incluyen nombres de personas, empresas, marcas y productos. Todos estos nombres son ficticios y cualquier similitud con nombres y direcciones utilizados por una empresa real es totalmente fortuita.

#### LICENCIA DE COPYRIGHT:

Este manual puede contener programas de aplicaciones de ejemplo escritos en lenguaje fuente, que muestran técnicas de programación en diversas plataformas operativas. Puede copiar, modificar y distribuir estos programas de ejemplo como desee, sin pago alguno a IBM con la intención de desarrollar, utilizar, comercializar o distribuir programas de aplicaciones de acuerdo con la interfaz de programación de aplicaciones correspondiente a la plataforma operativa para la que están escritos los programas de ejemplo. Estos ejemplos no se han probado exhaustivamente bajo todas las condiciones. Por lo tanto, IBM no puede asegurar ni implicar la fiabilidad, utilidad o función de estos programas.

Cada copia o parte de estos programas de ejemplo o cualquier trabajo derivado debe incluir una nota de copyright como la siguiente:

© (*nombre de la empresa*) (*año*). Partes de este código proceden de programas de ejemplo de IBM Corp. © Copyright IBM Corp. *\_entre el o los años\_*. Reservados todos los derechos.

### **Marcas registradas**

Los siguientes términos son marcas registradas de International Business Machines Corporation en los EE.UU. y/o en otros países.

pureXML	VisualAge
MQSeries	OpenPower
DB2	REXX
System z9	AIX
AISPO	System z
POWER	Encina
WebSphere	OS/390
DB2 Connect	DB2 Universal Database
TXSeries	Redbooks
z/OS	developerWorks
System i	PowerPC
AS/400	CICS
IBM	SQL/400
zSeries	Lotus
Rational	HACMP
Tivoli	MVS
OS/400	eServer
Approach	ibm.com
pSeries	iSeries

Los siguientes términos son marcas registradas de otras empresas.

- Linux es una marca registrada de Linus Torvalds en los Estados Unidos y/o en otros países.
- Java y todas las marcas comerciales basadas en Java son marcas comerciales de Sun Microsystems, Inc. en los Estados Unidos y/o en otros países.
- UNIX es una marca registrada de The Open Group en los Estados Unidos y/o en otros países.
- Intel Xeon, Itanium y Pentium e Intel son marcas comerciales de Intel Corporation o de sus subsidiarias en los Estados Unidos y/o en otros países.
- Microsoft y Windows son marcas comerciales de Microsoft Corporation, Inc. en los Estados Unidos y/o en otros países.

Otros nombres de empresas, productos o servicios, pueden ser marcas registradas o marcas de servicio de otras empresas.

---

# Índice

## A

- actualizaciones
  - Centro de información de DB2 48
- avisos 53
- ayuda
  - idioma de configuración 47
  - sentencias SQL 47

## B

- bases de datos
  - conexiones
    - Perl 37

## C

- Centro de información de DB2
  - actualización 48
  - idiomas 47
  - versiones 47
  - visualización en distintos idiomas 47

## D

- desarrollo de aplicaciones
  - Perl
    - creación de aplicaciones 40
- determinación de problemas
  - guías de aprendizaje 50
  - información disponible 50
- diseño de aplicación
  - ejemplo de Perl 39
  - prototipo en Perl 37
- documentación
  - copia impresa 43
  - PDF 43
  - términos y condiciones de uso 51
  - visión general 43

## E

- ejemplos
  - programa Perl 39
- errores
  - Perl 39

## G

- guías de aprendizaje
  - determinación de problemas 50
  - resolución de problemas 50
  - Visual Explain 50

## M

- manuales
  - copia impresa
    - pedido 46

- marcadores de parámetros
  - Perl 38

## P

- pedido de manuales de DB2 46
- Perl
  - aplicaciones 40
    - ejemplo 39
  - conexión con bases de datos 37
  - controladores 37
  - devolución de datos 38
  - ejemplos
    - aplicación 39
  - marcadores de parámetros 38
  - restricciones 39
  - SQLCODE 39
  - SQLSTATE 39
  - visión general 37
- PHP
  - configuración del entorno
    - Linux 5
    - Windows 5
  - extensión IBM\_DB2
    - conexión con una base de datos 9
    - emisión de sentencias SQL (procedimiento) 12, 13
    - expresiones XQuery 11
    - inserción de grandes objetos 14
    - invocación de procedimientos almacenados (parámetros IN o OUT) 19
    - invocación de procedimientos almacenados (varios conjuntos de resultados) 20
    - manejo de errores 18
    - mensajes de aviso 18
    - preparación de sentencias SQL 13
    - recuperación de columnas de resultados 15
    - recuperación de filas de resultados 16
    - recuperación de metadatos de la base de datos 9
    - recuperación de objetos grandes 16
    - transacciones 17
  - extensión PDO\_IBM
    - conexión con una base de datos 23
    - emisión de sentencias SQL (procedimiento) 24, 25
    - inserción de grandes objetos 26
    - invocación de procedimientos almacenados (parámetros OUT o INOUT; parámetros IN o OUT) 31
    - invocación de procedimientos almacenados (varios conjuntos de resultados) 32
    - manejo de errores 31
    - mensajes de aviso 31
    - preparación de sentencias SQL 25
    - recuperación de columnas de resultados 27
    - recuperación de filas de resultados 28
    - recuperación de objetos grandes 29
    - transacciones 30
  - visión general 3

## R

- recuperación de datos
  - Perl 38
- resolución de problemas
  - guías de aprendizaje 50
  - información en línea 50

## S

- sentencias SQL
  - visualización de la ayuda 47
- SQL dinámico
  - soporte de Perl 37
- SQL estático
  - Perl, no soportado 39

## T

- términos y condiciones
  - uso de publicaciones 51

## V

- variables del lenguaje principal
  - Perl 38
- Visual Explain
  - guía de aprendizaje 50







SC11-3502-01



Spine information:

DB2 Versión 9.5 para Linux, UNIX y Windows

Desarrollo de aplicaciones Perl y PHP

