



Hola invitado

[Inicio](#) [Buscar](#) [Enlaces](#) [Descargas](#) [Enviar Noticia](#)

21 Mar, 2007 - 04:31

## Menú principal

- Inicio
- Temas
- Secciones
- Descargas
- Enlaces

- Indice
- Conferencias

- Enviar noticia
- Búsquedas
- Usuarios
- Preferencias
- Top 10
- P+F [FAQ]
- RSS

## Anuncios

### [Anuncios Google](#)

## VFP y MySQL

Enviado por rfynn en Jueves, 08 Junio, 2006



Como conectarse a MySQL desde VFP. Acceder a tablas remotas, creando sentencias SQL simples y complejas. Pasando parámetros a MySQL y ejecutar procedimientos almacenados. Uso de vistas remotas desde VFP a MySQL.

lecturas 8559

## VFP y MySQL



Por Ricardo Fynn Reissig

### Crear una conexión a MySQL

Este ejemplo muestra como conectarse a una base de datos MySQL y trabajar en ella mediante SQL pass through.

Primero creamos el string de conexión a la misma, aquí damos dos ejemplos, con la base de datos en nuestro disco local y con una base de datos remota y en un puerto diferente.

En este ejemplo estamos usando una práctica llamada DSN-less es decir no usamos el administrador de orígenes de ODBC para conectarnos.

Es una práctica bastante usada y nos independizamos si el usuario creo la conexión ODBC en su puesto de trabajo o no.

Para crear una conexión a MySQL deberá tener instalado el driver ODBC correspondiente, este se puede descargar del siguiente link: <http://www.mysql.com/products/connector/odbc>

```
lcStringCnxRemoto = "DRIVER={MySQL ODBC 3.51 Driver};" + ;
                  "SERVER=200.1.1.1;" + ;
                  "PORT=3333;" + ;
                  "UID=booking;" + ;
                  "PWD=booking;" + ;
                  "DATABASE=booking;" + ;
                  "OPTIONS=131329;"
```

```
lcStringCnxLocal = "DRIVER={MySQL ODBC 3.51 Driver};" + ;
                  "SERVER=localhost;" + ;
                  "UID=root;" + ;
                  "PWD=clave;" + ;
                  "DATABASE=booking;" + ;
                  "OPTIONS=131329;"
```

```
SQLSETPROP(0,"DispLogin", 3)
lnHandle = SQLSTRINGCONNECT(lcStringCnxLocal)
```

### Interpretando el string de conexión

- **DRIVER** indica que estamos usando el driver MySQL instalado. En este ejemplo la versión 3.51
- **SERVER** indica donde está la base de datos MySQL, presentamos dos ejemplos para trabajar localmente y remoto. En el caso que sea remotopodemos especificar un número IP o el nombre de un servidor.
- **PORT** especifica donde está instalado el MySQL, muchas veces los administradores de red cambian este puerto (que por defecto es el 3306 por otro).
- **UID** es el nombre de login con el cual accederemos.
- **PWD** es nuestra clave.
- **DATABASE** es el nombre del schema de MySQL en que trabajaremos. en este ejemplo el schema booking.
- **OPTIONS** esta opción especifica como MySQL debería trabajar. Si no se usa nos podemos conectar sin problemas. Se utiliza en caso particulares.

En el siguiente link encontraras más información: <http://dev.mysql.com/doc/refman/5.0/en/connection-parameters.html>

### Accediendo a los datos

Vamos a probar nuestro string de conexión.

```
SQLSETPROP(0,"DispLogin", 3)
lnHandle = SQLSTRINGCONNECT(lcStringCnxLocal)
IF lnHandle > 0
  cmd = SQLEXP( lnHandle,"select QuantityRooms from roomtype","cur_roomtype")
  IF cmd > 0
```

[Anuncios Goooooogle](#)

### [Escritorio Remoto](#)

Acceso remoto a PC gratuito!  
Soluciones de soporte a distancia  
es.LogMeIn.com

### [PostgreSQL Solutions](#)

worldwide Training Courses, Support Software Development, Consulting  
www.postgresql.at

### [SQL injection scanner](#)

Find out if your web site is vulnerable to SQL injection attacks  
www.acunetix.com/sql-inje

### [MySQL Tool](#)

Monitor MySQL Performance Manage LAMP Apps. Download Now!  
appmanager.com/MySQL

[Anunciarse en este sitio](#)

### En línea

Actualmente hay 2 usuarios registrados en línea.

Puedes [loguearte o crear una cuenta nueva aqui](#).

```
BROWSE
ELSE
  AERROR(laErr)
  MESSAGEBOX("No se pudo conectar a MySQL. Error: " + CHR(13) + laErr[2])
ENDIF
USE IN cur_roomtype
SQLDISCONNECT(lnHandle)
ELSE
  AERROR(laErr)
  MESSAGEBOX("No se pudo conectar a MySQL. Error: " + CHR(13) + laErr[2])
ENDIF
```

Si nuestra conexión fue satisfactoria, la variable **lnHandle** fue mayor que cero, comenzamos a hacer operaciones SQL contra la base de datos.

En este ejemplo simplemente hacemos un select que nos devuelve la columna QuantityRooms de la tabla RoomType. El resultado lo guardamos en un cursor llamado cur\_roomtype.

#### Creando sentencias SQL más complejas

El comando SQLEXEC() tiene como segundo parámetro la sentencia SQL. Esta, si es muy compleja la podemos guardar en una variable y si esta es mayor de 255 caracteres (límite de VFP para las variables de tipo string) podemos usar el comando TEXT y además queda más elegante en nuestro código

```
TEXT TO lcSQLcommand NOSHOW
  SELECT Inventory.inventoryid, Inventory.date, Inventory.roomtypeid,
  Inventory.sold, Roomtype.descrip, Roomtype.rooms
  FROM inventory Inventory
  INNER JOIN roomtype Roomtype
  ON Inventory.roomtypeid = Roomtype.roomTypeID
  ORDER BY Inventory.date DESC
ENDTEXT
cmd = SQLEXEC(lnHandle,lcSQLcommand)
```

Nótese que no usa la coma para separar concatenar los comandos. Dentro del bloque **TEXT ... ENDTEXT** podemos escribir como queramos.

**Un punto muy importante es que si nuestra base de datos MySQL reside en un server Linux o Unix esta será case-sensitive, o sea, las distinguirá mayúsculas de minúsculas. No es lo mismo SELECT roomtype que SELECT RoomType. Esto generará un error dependiendo como hayamos creado nuestro schema.**

#### Pasando parámetros a nuestra consulta

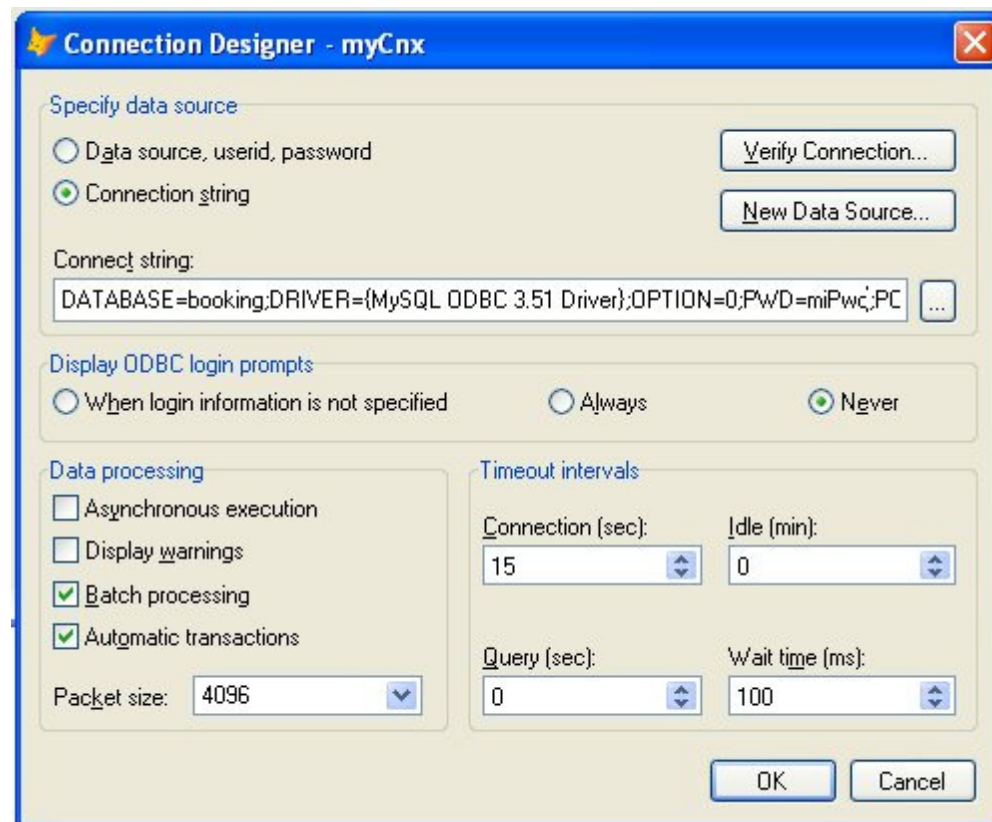
La manera más sencilla de pasar parámetros es definiendo una variable, asignándole un valor previo a la ejecución de la consulta y asignándole un signo de interrogación delante.

```
nValor = 12
cmd = SQLEXEC(lnHandle,"select QuantityRooms from roomtype where roomtypeid = ?nValor","cur_roomtype")
```

#### Llamando un procedimiento almacenado

```
SQLEXEC(lnHandle, "call myStoreProcedure(@param1)")
```

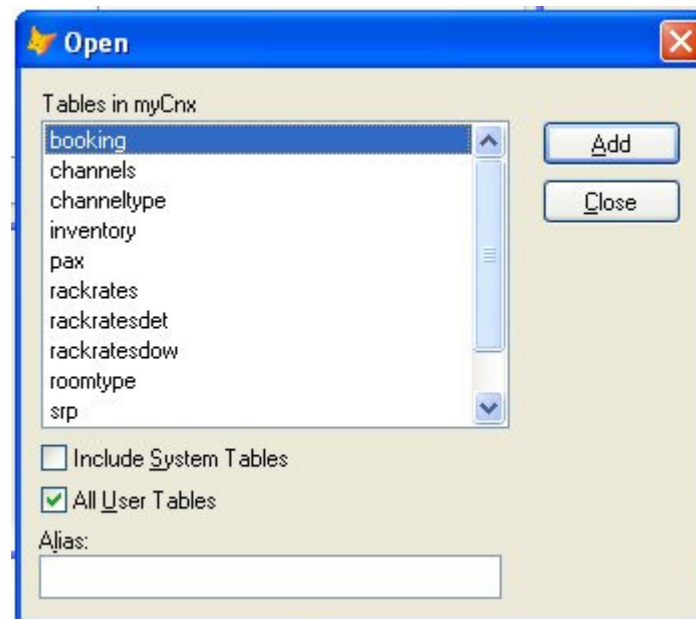
#### Usando vistas remotas a MySQL



El primer paso es crear la conexión en nuestro contenedor de base de base datos de VFP.

Como string de conexión usamos el mismo con el que comenzamos estos ejemplos.

Verificamos que la conexión sea correcta y comenzamos a crear una vista remota.



Una vez que seleccionamos en nuestro contenedor de VFP crear una vista remota, seleccionamos la conexión creada en el paso anterior.

Automáticamente VFP nos pedirá que agreguemos tablas a al diseñador de vistas.

Un punto importante es que debe estar marcado el checkbox "All User Tables", de lo contrario el driver de MySQL nos devolverá un error como que la tabla seleccionada no existe.

#### Algunas consideraciones importantes

Cuando diseñemos en MySQL nuestro schema de datos, es importante crear un usuario que no sea **root** para acceder al mismo y darle los privilegios necesarios para trabajar sobre el schema.

**root** es el usuario con máximos privilegios sobre la base de datos (ídem que el usuario **sa** en SQL Server) y no es recomendable trabajar con él y menos si se accede en forma remota.

Si nuestro motor MySQL reside en un server Linux (o Unix) tener consideración con los nombres de las tablas y columnas, estos distinguen mayúsculas de las minúsculas.

No existe una estandarización en el lenguaje de los motores de base de datos, por lo cual si en SQL Server usaban la cláusula SELECT TOP 10 FROM miTabla ... para obtener solo 10 registros de una consulta, esta generará un error en MySQL.

MySQL usa la cláusula LIMIT 10. Es muy importante que lean la ayuda sobre cláusulas y funciones y estas no queden "hard-coded" en el código y puedan parametrizarse en el sistema, así si pasan de MySQL a ORACLE o a SQL Server, cambien la capa de datos y no deban tocar su código.



VFP y MySQL | [Entrar/Crear una cuenta](#) | 0 Comentarios

Mostrar  Orden

Los comentarios son propiedad de sus respectivos autores.  
No somos responsables de su contenido.

#### [Advanced Query Tool](#)

Cross-database query and admin tool.  
Powerful, fast, inexpensive.

[Anuncios Goooooogle](#)

#### [Foxpro, Clipper and dBase](#)

Anything about Foxpro, Clipper, dBase: all  
xBase info is here.

[Anunciarse en este sitio](#)

Todas las marcas y los logos utilizados en este sitio son propiedad de sus respectivos dueños.  
Los artículos, noticias y comentarios son propiedad y responsabilidad de sus respectivos autores.  
Copyright © 2000-2007 PortalFox. Todos los derechos reservados.