

Query Management Facility



# Iniciación a QMF para Windows

*Versión 7*



Query Management Facility



# Iniciación a QMF para Windows

*Versión 7*

**Nota:**

Antes de utilizar esta información y el producto al que da soporte, lea la información general que figura bajo el apartado "Apéndice. Avisos" en la página 125.

**Primera Edición (Setiembre de 2000)**

Este manual es la traducción del original inglés *Query Management Facility - Getting Started with QMF for Windows Version 7 Release 1 (SC27-0723-00)*

Esta edición se aplica a Query Management Facility para Windows, una característica de la Versión 7 de DB2 Universal Database Server para OS/390 (DB2 UDB para OS/390), 5675-DB2, y de Query Management Facility, una característica de la Versión 7 de DATABASE 2 Server para VM y VSE (DB2 para VM y VSE), 5697-F42, y de Query Management Facility para Windows para AS/400, 5697-G24, y de Query Management Facility para Windows para DB2 Workstation Databases, 5697-G22, y de DB2 Warehouse Manager, 5648-D35, y de DB2 Warehouse Manager para AS/400, 5697-G23, y a todos los releases y modificaciones subsiguientes hasta que se indique lo contrario en nuevas ediciones.

Esta edición sustituye y deja obsoleta a la edición anterior, SC26-9582-02. Los cambios técnicos mencionados en esta edición aparecen identificados mediante una barra vertical a la izquierda de cada cambio. Los cambios editoriales que carecen de relevancia técnica no aparecen indicados.

© Copyright International Business Machines Corporation 1997, 2000. Reservados todos los derechos.

# Contenido

## La biblioteca de QMF . . . . . vii

### Capítulo 1. Introducción . . . . . 1

Servidores de bases de datos . . . . .	1
Familia de bases de datos DB2 . . . . .	1
Nombre de usuario frente a nombre técnico . . . . .	1
Establecimiento del nombre de servidor . . . . .	1
Seguridad de las bases de datos . . . . .	2
Inicio de sesión . . . . .	2
Corrección de contraseñas . . . . .	3
Cambio de contraseñas . . . . .	3
Especificación de cuentas . . . . .	3
Gobierno . . . . .	4
Visualización de límites de recursos . . . . .	4
Establecimiento del propio límite de filas . . . . .	4
Personalización de la barra de herramientas . . . . .	5
Adición de botones a la barra de herramientas . . . . .	5
Cómo mover botones en la barra de herramientas . . . . .	5
Eliminación de botones de la barra de herramientas . . . . .	6

### Capítulo 2. Cómo trabajar con consultas de SQL . . . . . 7

Consultas de SQL . . . . .	7
Creación de consultas de SQL nuevas . . . . .	7
Ejecución de consultas de SQL en un servidor de bases de datos . . . . .	7
Conmutación entre la vista de resultados y la vista de SQL . . . . .	7
Cómo trabajar con fuentes . . . . .	8
Selección de la fuente de visualización de la consulta . . . . .	8
Múltiples consultas . . . . .	8
Visualización simultánea de múltiples consultas . . . . .	8
Describir consultas . . . . .	8
Creación de consultas de SQL nuevas . . . . .	9
Variables de sustitución en consultas de SQL . . . . .	10
Ejecución de consultas de SQL con variables de sustitución . . . . .	10
Cómo guardar y abrir consultas de SQL . . . . .	10
Cómo guardar consultas de SQL en archivos . . . . .	11
Cómo abrir archivos de consultas de SQL guardados . . . . .	11
Cómo guardar consultas de SQL en el servidor de bases de datos . . . . .	11
Cómo abrir consultas de SQL guardadas en el servidor de bases de datos . . . . .	11
Impresión de consultas de SQL . . . . .	12
Presentación preliminar de una consulta . . . . .	12
Impresión de consultas de SQL . . . . .	12

### Capítulo 3. Cómo trabajar con consultas solicitud . . . . . 13

Creación de consultas simples . . . . .	13
---	----

Cómo abrir consultas solicitud nuevas . . . . .	13
Botones de acción de consulta solicitud . . . . .	13
Adición de tablas a las consultas solicitud . . . . .	14
Ejecución de consultas solicitud . . . . .	14
Creación de consultas complejas . . . . .	14
Adición de columnas a consultas solicitud . . . . .	14
Utilización de condiciones de clasificación . . . . .	15
Adición de condiciones de clasificación . . . . .	15
Utilización de condiciones de fila . . . . .	16
Adición de condiciones de fila . . . . .	16
Utilización de múltiples tablas en consultas solicitud . . . . .	17
Creación de condiciones de unión de consulta solicitud . . . . .	17
Utilización de SQL en consultas solicitud . . . . .	18
Visualización de SQL para consultas solicitud . . . . .	18
Conversión de consultas solicitud en SQL . . . . .	18
Utilización de variables de sustitución en consultas solicitud . . . . .	18
Cómo guardar consultas solicitud . . . . .	18
Cómo guardar consultas solicitud en archivos . . . . .	19
Cómo abrir archivos de consultas solicitud guardados . . . . .	19
Cómo guardar consultas solicitud en el servidor de bases de datos . . . . .	19
Cómo abrir consultas solicitud guardadas en el servidor de bases de datos . . . . .	19
Impresión de consultas solicitud . . . . .	20
Presentación preliminar de consultas solicitud . . . . .	20

### Capítulo 4. Cómo trabajar con resultados de consultas . . . . . 21

Clasificación y ajuste de los resultados de una consulta . . . . .	21
Selección de columnas y filas . . . . .	21
Cambio de tamaño de las columnas y filas . . . . .	21
Ajuste automático de columnas y filas . . . . .	21
Clasificación de resultados de consulta . . . . .	22
Reordenación de columnas . . . . .	22
Cómo aplicar formato a resultados de consulta . . . . .	22
Selección de la fuente de visualización de resultados de consulta . . . . .	22
Cómo aplicar formato a resultados de consulta numéricos . . . . .	23
Conversión de los valores de formato de los resultados de consulta en un formato . . . . .	23
Cómo agrupar y agregar resultados de consulta . . . . .	23
Agrupación de resultados de consulta . . . . .	23
Cómo resumir resultados de consulta . . . . .	23
Cómo guardar resultados de consulta y el formato . . . . .	23
Cómo guardar resultados de consulta como una tabla . . . . .	23
Cómo guardar resultados de consulta en archivos . . . . .	24
Impresión de resultados de consulta . . . . .	24
Presentación preliminar de resultados de consulta . . . . .	24

Impresión de resultados de consulta . . . . . 24

## Capítulo 5. Cómo trabajar con informes 25

Formatos . . . . .	25
Interpretación de los formatos . . . . .	25
Cómo producir un informe utilizando un formato . . . . .	26
Edición de un formato. . . . .	26
Creación de un formato . . . . .	26
Paso 1: Crear un formato . . . . .	26
Paso 2: Cambiar el orden de las columnas . . . . .	27
Paso 3: Cambiar los encabezamientos de columna . . . . .	27
Paso 4: Cambiar el formato de columna . . . . .	27
Paso 5: Añadir información de resumen . . . . .	27
Paso 6: Añadir encabezamientos y pies de página . . . . .	28
Cómo guardar formatos . . . . .	28
Cómo guardar un formato en un archivo . . . . .	28
Cómo abrir un archivo de formato guardado . . . . .	28
Cómo guardar formatos en el servidor de bases de datos . . . . .	28
Cómo abrir formatos guardados en el servidor de bases de datos . . . . .	29
Impresión de informes. . . . .	29
Exportación de informes . . . . .	30

## Capítulo 6. Cómo trabajar con procedimientos . . . . . 31

Ejecución de procedimientos. . . . .	31
Creación de un procedimiento lineal nuevo. . . . .	31
Creación de un procedimiento con lógica nuevo . . . . .	31
Ejecución de un procedimiento en un servidor de bases de datos . . . . .	31
Cómo guardar procedimientos . . . . .	32
Cómo guardar procedimientos en archivos . . . . .	32
Cómo abrir un archivo de procedimiento guardado . . . . .	32
Cómo guardar un procedimiento en el servidor de bases de datos . . . . .	32
Cómo abrir procedimientos guardados en el servidor de bases de datos . . . . .	33
Impresión de procedimientos . . . . .	33
Presentación preliminar de un procedimiento . . . . .	33
Impresión de un procedimiento. . . . .	33

## Capítulo 7. Cómo trabajar con listas . . . 35

Objetos . . . . .	35
Listado de objetos . . . . .	35
Mandatos de la ventana Lista . . . . .	36
Creación de listas . . . . .	36
Cómo añadir objetos a las listas . . . . .	36
Eliminación de objetos de las listas . . . . .	36
Guardar listas en archivos . . . . .	37
Cómo abrir archivos de lista guardados . . . . .	37

## Capítulo 8. Cómo trabajar con archivos de trabajos . . . . . 39

Archivos de trabajos . . . . .	39
Creación de archivos de trabajos . . . . .	39
Ejecución de archivos de trabajos . . . . .	39
Ajuste automático de columnas y filas . . . . .	39
Clasificación de resultados de consulta . . . . .	39

Reordenación de columnas . . . . .	40
Cómo aplicar formato a resultados de consulta . . . . .	40
Selección de la fuente de visualización de resultados de consulta. . . . .	40
Cómo aplicar formato a resultados de consulta numéricos . . . . .	40
Conversión del formato de los resultados de consulta en un formato . . . . .	41
Cómo agrupar y agregar resultados de consulta . . . . .	41
Agrupación de resultados de consulta . . . . .	41
Cómo resumir resultados de consulta. . . . .	41
Cómo guardar resultados de consulta y el formato . . . . .	41
Cómo guardar resultados de consulta como una tabla. . . . .	41
Cómo guardar resultados de consulta en archivos . . . . .	41
Impresión de resultados de consulta . . . . .	42
Presentación preliminar de resultados de consulta . . . . .	42
Impresión de resultados de consulta . . . . .	42

## Capítulo 9. Cómo trabajar con consultas estáticas . . . . . 43

Consultas estáticas . . . . .	43
Creación de consultas estáticas . . . . .	43
Sustitución de variables de sustitución por variables del sistema principal . . . . .	44
Ejecución de un consulta estática . . . . .	45

## Capítulo 10. Cómo trabajar con el Editor de tablas . . . . . 47

Editor de tablas . . . . .	47
Búsqueda de filas utilizando el editor de tablas . . . . .	47
Adición de una fila. . . . .	48
Cambio de una fila . . . . .	48
Supresión de una fila . . . . .	48
Edición de tablas desde la vista de resultados de consulta . . . . .	48
Supresión de una fila desde la vista de resultados de consulta . . . . .	49
Actualización de columnas desde la vista de resultados de consulta . . . . .	49
DB2 Forms . . . . .	49

## Capítulo 11. Distribución de datos . . . 51

Exportación de datos . . . . .	51
Exportación de datos a archivos . . . . .	51
Importación de datos . . . . .	52
Cómo guardar datos en un servidor de bases de datos . . . . .	52
Utilización del mandato Enviar a . . . . .	53
Utilización de la macro automática (Complementos) de Microsoft Excel . . . . .	53
Utilización de aplicaciones de ejemplo . . . . .	54

## Capítulo 12. Utilización del Centro de informes de QMF . . . . . 55

Iniciación rápida del Centro de informes de QMF . . . . .	55
Ventana del Centro de informes de QMF . . . . .	55
Conexión al servidor . . . . .	56
Cómo trabajar con Informes y Objetos . . . . .	57

Ejecución de informes . . . . .	57
Cómo trabajar con carpetas y favoritos . . . . .	58
Cómo añadir informes a favoritos . . . . .	59

## Capítulo 13. Utilización de la API de QMF para Windows . . . . . 61

Cómo controlar QMF para Windows mediante la API . . . . .	61
Bloqueo de llamadas . . . . .	61
Conexión a la base de datos . . . . .	62
Consulta de la API . . . . .	62
AddDecimalHostVariable() . . . . .	62
AddHostVariable() . . . . .	63
BindDecimalHostVariable() . . . . .	64
BindHostVariable() . . . . .	64
BindSection() . . . . .	65
CancelBind() . . . . .	66
ChangePassword() . . . . .	66
ClearList() . . . . .	67
Close() . . . . .	67
Commit() . . . . .	68
CompleteQuery() . . . . .	68
CopyToClipboard() . . . . .	68
DeleteQMFObject() . . . . .	69
EndBind() . . . . .	70
Execute() . . . . .	70
ExecuteEx() . . . . .	70
ExecuteStored Procedure() . . . . .	71
ExecuteStored ProcedureEx() . . . . .	72
Export() . . . . .	73
ExportForm() . . . . .	75
ExportReport() . . . . .	76
FastSaveData() . . . . .	77
FetchNextRow() . . . . .	78
FetchNextRowEx() . . . . .	79
FetchNextRows() . . . . .	79
FetchNextRowsEx() . . . . .	80
FlushQMFCache() . . . . .	81
GetColumnCount() . . . . .	81
GetColumnDataValue() . . . . .	82
GetColumnHeader() . . . . .	82
GetColumnHeaderEx() . . . . .	82
GetColumnHeadings() . . . . .	83
GetColumnValue() . . . . .	84
GetColumnValueEx() . . . . .	84
GetDefaultServerName() . . . . .	85
GetGlobalVariable() . . . . .	85
GetHostVariableNames() . . . . .	85
GetHostVariableTypes() . . . . .	86
GetLastErrorString() . . . . .	86
GetLastErrorType() . . . . .	87
GetLastSQLCode() . . . . .	88
GetLastSQLError() . . . . .	88
GetLastSQLState() . . . . .	89

GetOption() . . . . .	89
GetOptionEx() . . . . .	91
GetProcText() . . . . .	91
GetProcVariables() . . . . .	91
GetQMFObjectInfo() . . . . .	92
GetQMFObjectInfoEx() . . . . .	94
GetQMFObjectList() . . . . .	95
GetQMFObjectListEx() . . . . .	96
GetQMFPProcText() . . . . .	97
GetQMFPQueryText() . . . . .	97
GetQueryText() . . . . .	98
GetQueryVerb() . . . . .	98
GetResourceLimit() . . . . .	98
GetResourceLimitEx() . . . . .	102
GetRowCount() . . . . .	102
GetServerList() . . . . .	103
GetServerListEx() . . . . .	104
GetStoredProcedureResultSets() . . . . .	104
GetVariables() . . . . .	105
GetVariablesEx() . . . . .	105
InitializeProc() . . . . .	106
InitializeQuery() . . . . .	106
InitializeServer() . . . . .	107
InitializeStaticQuery() . . . . .	108
IsStatic() . . . . .	108
Open() . . . . .	109
Prepare() . . . . .	110
PrintReport() . . . . .	110
ReinitializeServer() . . . . .	110
Rollback() . . . . .	111
RunProc() . . . . .	111
SaveData() . . . . .	111
SaveQMFPProc() . . . . .	113
SaveQMFPQuery() . . . . .	114
SetBindOption() . . . . .	114
SetBindOwner() . . . . .	116
SetBusyWindowButton() . . . . .	116
SetBusyWindowMessage() . . . . .	117
SetBusyWindowMode() . . . . .	117
SetBusyWindowTitle() . . . . .	118
SetGlobalVariable() . . . . .	118
SetHostVariable() . . . . .	119
SetOption() . . . . .	119
SetParent() . . . . .	120
SetProcVariable() . . . . .	121
SetVariable() . . . . .	121
ShowBusyWindow() . . . . .	122
StartBind() . . . . .	122

## Apéndice. Avisos. . . . . 125

Marcas registradas. . . . .	127
-----------------------------	-----

## Índice. . . . . 129



---

## La biblioteca de QMF

Los manuales pueden solicitarse a través de un representante de IBM.

Evaluación	 <p>Introducción a QMF GC10-3530</p>			
Instalación, planificación, administración y diagnóstico	 <p>Installing and Managing QMF on OS/390 GC26-9575</p>	 <p>Installing and Managing QMF on VM/ESA GC26-9573</p>	 <p>Installing and Managing QMF on VSE/ESA GC26-9574</p>	 <p>Installing and Managing QMF for Windows GC26-9583</p>
	 <p>QMF Messages and Codes GC26-9580</p>	 <p>QMF High Performance Option User's Guide for OS/390 SC26-9581</p>		
Utilización	 <p>Utilización de QMF SC10-3532</p>	 <p>Consulta de QMF SC10-3531</p>	 <p>Iniciación a QMF para Windows SC10-3533</p>	
Programación de aplicaciones	 <p>Desarrollo de aplicaciones de QMF SC10-3137</p>			
Bibliotecas en línea	 <p>SK2T-0730 OS/390, VM y VSE</p>	 <p>SK2T-6700 Sólo OS/390</p>	 <p>SK2T-2067 Sólo VM</p>	 <p>SK2T-0060 Sólo VSE</p>



---

## Capítulo 1. Introducción

Este capítulo facilita una visión general de QMF para Windows y explica alguna de las tareas básicas sobre el uso de QMF para Windows.

---

### Servidores de bases de datos

Las consultas, formatos, procedimientos y tablas se ejecutan y se guardan en un servidor de bases de datos.

#### Familia de bases de datos DB2

QMF para Windows puede conectarse a una amplia gama de bases de datos DB2.

- DB2 UDB para OS390, DB2 para OS390 y DB2 para MVS
- DB2 Server para VSE & VM y SQLDS
- DB2 Universal Database y DB2 Common Server
- DB2 Parallel Edition
- DataJoiner

La licencia de QMF para Windows determina qué productos de la familia DB2 puede instalar y conectar con su copia de QMF para Windows.

#### Nombre de usuario frente a nombre técnico

Las distintas versiones y tipos de DB2 hacen referencia a una base de datos mediante un nombre RDB, un nombre de ubicación o algún otro nombre técnico.

Gracias a QMF para Windows, el administrador puede asignar un nombre a la base de datos que al usuario le resulte sencillo de recordar, por ejemplo, Base de datos de Compras en lugar de DB2P\_01\_PURCH.

QMF para Windows hace referencia a un servidor de bases de datos o a una base de datos DB2 denominándole "servidor."

#### Establecimiento del nombre de servidor

Para poder consultar una base de datos, QMF para Windows necesita saber dónde está almacenada la base de datos.

1. En el menú **Archivo**, seleccione **Nueva consulta de SQL**. Se abrirá un nuevo documento de consulta de SQL.

2. En el menú **Consulta**, seleccione **Establecer servidor**. Aparecerá el recuadro de diálogo Establecer servidor.



3. En la lista de servidores disponibles, seleccione el que desea consultar y pulse en **Aceptar**. Cuando empiece la próxima sesión de QMF para Windows, QMF para Windows le volverá a conectar automáticamente al mismo servidor.

---

## Seguridad de las bases de datos

Para poderse conectar a un servidor, debe proporcionar un ID de usuario y una contraseña.

### Inicio de sesión

Debe especificar un ID de usuario y una contraseña válidos para el servidor de bases de datos al que está intentando acceder. La contraseña y el ID de usuario del servidor de bases de datos no tienen por qué coincidir necesariamente con la contraseña y el ID de usuario locales o de red.

Si está ejecutando Windows, tiene la opción de recordar contraseñas de servidor de una sesión de QMF para Windows a otra. Si actualmente ha iniciado la sesión en Windows, el recuadro de diálogo Establecer información de usuario visualiza un recuadro de selección adicional llamado **¿Desea recordar esta contraseña?** Si marca este recuadro de selección, la contraseña entrada para dicho servidor se almacenará en la lista de contraseñas de Windows. Siempre que inicie la sesión en Windows, QMF para Windows podrá recuperar automáticamente dicha contraseña para que no se le solicite. Si no ha iniciado la sesión cuando ejecuta QMF para Windows, o si ha iniciado la sesión como un usuario diferente, QMF para Windows le solicitará un ID de usuario y una contraseña.

**Nota:** Si el usuario elige guardar una contraseña, cualquier persona que pueda conectarse a la cuenta de Windows de dicho usuario podrá acceder a los servidores de bases de datos con el ID de usuario y la contraseña (de servidor) de ese usuario.

1. En el menú **Consulta**, seleccione **Establecer información del usuario**. Se abrirá el recuadro de diálogo Establecer información del usuario.

Establecer información del usuario

Especificar información del usuario para: DOLPHIN

ID de usuario:

Contraseña:

Nueva contraseña:

Confirmar nueva contraseña:

¿Desea recordar esta contraseña?

Cuenta:

Aceptar

Cancelar

<< Ocultar

2. Entre el ID de usuario y la contraseña en los campos apropiados.

**Nota:** El ID de usuario y la contraseña son sensibles a las mayúsculas y minúsculas. Por ejemplo, si el ID de usuario o la contraseña están en mayúsculas, debe especificarlo en mayúsculas. Algunos tipos de servidores de bases de datos tienen en cuenta las mayúsculas y minúsculas en los ID de usuario y en las contraseñas, mientras que otros no lo tienen en cuenta.

3. Seleccione **¿Desea recordar esta contraseña?** si elige guardar el ID de usuario y la contraseña.
4. Pulse en **Aceptar**. QMF para Windows almacenará esta información durante la preparación para acceder al servidor.

## Corrección de contraseñas

Si ha entrado una contraseña incorrecta, puede corregir el error volviendo a abrir el recuadro de diálogo Establecer información del usuario.

1. En el menú **Consulta**, pulse en **Establecer información del usuario**. Se abrirá el recuadro de diálogo Establecer información del usuario.
2. Vuelva a escribir la contraseña y pulse en **Aceptar**. Se corregirá la contraseña.

## Cambio de contraseñas

Puede cambiar la contraseña en el servidor de bases de datos desde QMF para Windows. Actualmente, esta característica sólo está soportada en DB2 para OS390, Versión 5 y posterior.

1. En el menú **Consulta**, seleccione **Establecer información del usuario**. Se abrirá el recuadro de diálogo Establecer información del usuario.
2. Pulse en **Cambiar**. Aparecerán los campos **Nueva contraseña** y **Confirmar nueva contraseña**.
3. Escriba la contraseña nueva en los campos **Nueva contraseña** y **Confirmar nueva contraseña** y pulse en **Aceptar**. Se cambiará la contraseña del servidor de bases de datos.

## Especificación de cuentas

El servidor de bases de datos utiliza cuentas para efectuar un seguimiento del uso del sistema. Solicite al administrador de bases de datos que averigüe si el sistema utiliza cuentas.

1. En el menú **Consulta**, seleccione **Establecer información del usuario**. Se abrirá el recuadro de diálogo Establecer información del usuario.

2. En el campo **Cuenta**, escriba la cuenta que desea utilizar y pulse en **Aceptar**. QMF para Windows almacenará la información durante la preparación para acceder al servidor.

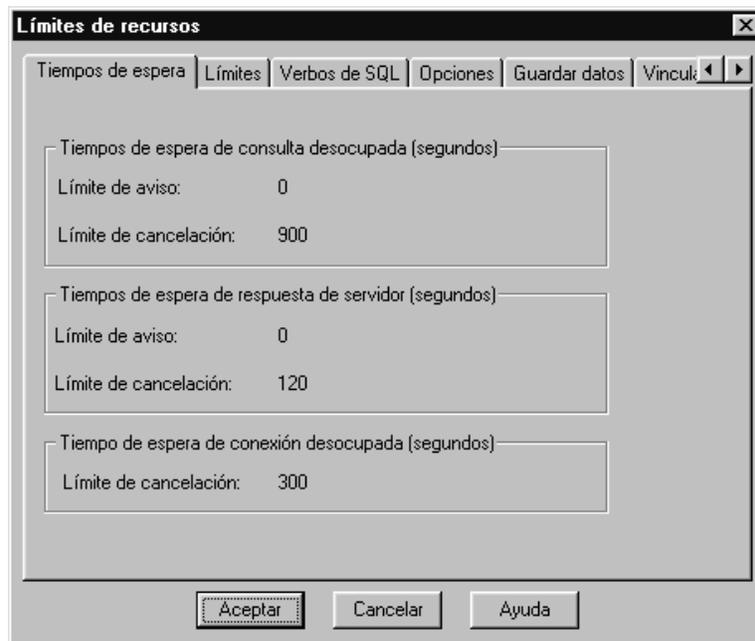
---

## Gobierno

El programa de gobierno de QMF para Windows siempre se ejecuta en segundo plano, supervisando el uso de los recursos del sistema y de las bases de datos. Este programa también establece límites en el tipo y tamaño de las consultas que se pueden ejecutar.

### Visualización de límites de recursos

En el menú **Ver**, seleccione **Límites de recursos**. Se abrirá el recuadro de diálogo Límites de recursos. Toda la información del recuadro de diálogo Límites de recursos es de sólo lectura. El administrador del sistema establece estos límites.



Los tipos de límites y controles que pueden estar en vigor son:

- Tiempos de espera
- Límites
- Verbos de SQL
- Opciones
- Guardar datos
- Vincular
- Seguimiento de objetos

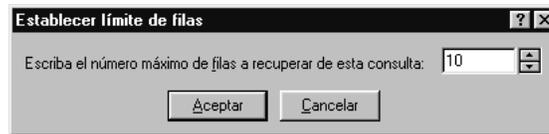
### Establecimiento del propio límite de filas

Entre el número máximo de filas que se van a recuperar para esta consulta. Una vez que se alcanza este límite, QMF para Windows cancela la consulta. El límite de filas máximo autorizado especificado en el grupo de límites de recursos tiene prioridad sobre este parámetro.

Entre 0 en este campo para especificar que no hay ningún límite.

Las filas que QMF para Windows ya ha recuperado excediendo este límite se retienen y están disponibles para su visualización.

1. En el menú **Consulta**, pulse en **Establecer límite de filas**. Se abrirá el recuadro de diálogo Establecer límite de filas.



2. Entre el número máximo de filas que desea que devuelva la consulta y pulse en **Aceptar**. La próxima vez que ejecute la consulta se aplicará el límite de filas.

---

## Personalización de la barra de herramientas

Puede personalizar la barra de herramientas para que se muestren sólo los botones que desea ver.

### Adición de botones a la barra de herramientas

Existe la posibilidad de añadir botones a la barra de herramientas actual de QMF para Windows. Estos botones representan funciones que quizá no necesiten todos los usuarios, pero que están disponibles para incluirlas en la barra de herramientas.

1. Efectúe una doble pulsación en el área gris que rodea la barra de herramientas. Se abrirá el recuadro de diálogo Personalizar la barra de herramientas.



2. En la columna **Botones disponibles**, seleccione el botón que desea añadir y pulse en **Agregar**. El botón se añadirá a la barra de herramientas.
3. Cuando haya terminado de añadir botones, pulse en **Cerrar**. Se cerrará el recuadro de diálogo y se añadirán los botones nuevos a la barra de herramientas.

### Cómo mover botones en la barra de herramientas

Existe la posibilidad de reorganizar los botones en la barra de herramientas de QMF para Windows.

1. Efectúe una doble pulsación en el área gris que rodea la barra de herramientas. Se abrirá el recuadro de diálogo Personalizar la barra de herramientas.
2. En la columna **Botones disponibles**, seleccione el botón que desea mover.
3. Utilice los botones **Subir** y **Bajar** para mover el botón dentro de la barra de herramientas.
4. Cuando haya terminado de mover botones, pulse en **Cerrar**. Se cerrará el recuadro de diálogo y aparecerá la barra de herramientas con los botones en sus nuevas ubicaciones.

## Eliminación de botones de la barra de herramientas

Existe la posibilidad de eliminar botones de la barra de herramientas de QMF para Windows.

1. Efectúe una doble pulsación en el área gris que rodea la barra de herramientas. Se abrirá el recuadro de diálogo Personalizar la barra de herramientas.
2. En la columna **Botones disponibles**, seleccione el botón que desea eliminar y pulse en **Quitar**. Se eliminará el botón de la barra de herramientas.
3. Cuando haya terminado de eliminar botones, pulse en **Cerrar**. Se cerrará el recuadro de diálogo y se eliminarán los botones de la barra de herramientas.

---

## Capítulo 2. Cómo trabajar con consultas de SQL

El Lenguaje de Consulta Estructurada (SQL) constituye la interfaz más básica entre un usuario y una base de datos. Las consultas se escriben en SQL y su proceso corre a cargo de la base de datos. Los usuarios pueden escribir las consultas de QMF para Windows en SQL, o bien crear consultas utilizando el método "señalar y pulsar".

---

### Consultas de SQL

Las consultas del Lenguaje de Consulta Estructurada requieren conocimientos sobre los mandatos y la sintaxis de SQL. Es preferible que los usuarios que no estén familiarizados con SQL creen Consultas solicitud.

#### Creación de consultas de SQL nuevas

Pulse en el botón **Nueva consulta de SQL** de la barra de herramientas.



Aparecerá un nuevo documento de consulta.

#### Ejecución de consultas de SQL en un servidor de bases de datos

1. Abra un documento de consulta nuevo y escriba una consulta, abra un archivo de consulta existente o abra una consulta de la base de datos.
2. Pulse en el botón **Ejecutar consulta** de la barra de herramientas.



Se ejecutará la consulta y se visualizarán los resultados.

#### Conmutación entre la vista de resultados y la vista de SQL

Puede examinar los resultados de una consulta o la sentencia SQL propiamente dicha.

En la vista de SQL de una consulta que se ha ejecutado, pulse en el botón **Ver resultados** de la barra de herramientas.



Se visualizarán los resultados de la consulta.

-o bien-

En la vista de resultados de una consulta, pulse en el botón **Ver SQL**.



Se visualizará la sentencia SQL.

---

## Cómo trabajar con fuentes

Es posible cambiar la fuente utilizada para visualizar consultas. La selección de fuentes dependerá de lo que se haya instalado en su sistema. Para obtener más información sobre cómo añadir fuentes, consulte el recurso de ayuda del sistema operativo.

**Nota:** Si guarda la consulta después de seleccionar una fuente nueva para visualizar la consulta, esta consulta siempre se visualizará con la fuente nueva.

### Selección de la fuente de visualización de la consulta

1. En la vista de SQL, pulse en **Establecer fuente** en el menú **Consulta**. Se abrirá el recuadro de diálogo Fuente.
2. Seleccione la fuente para visualizar el texto de la consulta y pulse en **Aceptar**. La consulta volverá a aparecer con la fuente nueva.

**Nota:** Pulse en **Establecer como valor por omisión** para utilizar la fuente seleccionada como fuente por omisión para todas las consultas nuevas.

---

## Múltiples consultas

Puede tener más de un documento de consulta abierto al mismo tiempo. También puede ejecutar más de una consulta a la vez. Esta función sirve para generar múltiples informes, o para cortar y pegar el texto de SQL de una consulta en otra.

### Visualización simultánea de múltiples consultas

1. Abra dos documentos de consulta, como mínimo.
2. En el menú **Ventana**, seleccione uno de los mandatos siguientes:

Mandato	Resultado
Cascada	Selecciónelo para visualizar consultas en una serie escalonada.
Mosaico horizontal	Selecciónelo para visualizar las ventanas de consulta apiladas verticalmente.
Mosaico vertical	Selecciónelo para visualizar las ventanas de consulta apiladas horizontalmente.

Las ventanas de consulta se organizan de acuerdo con la opción seleccionada.

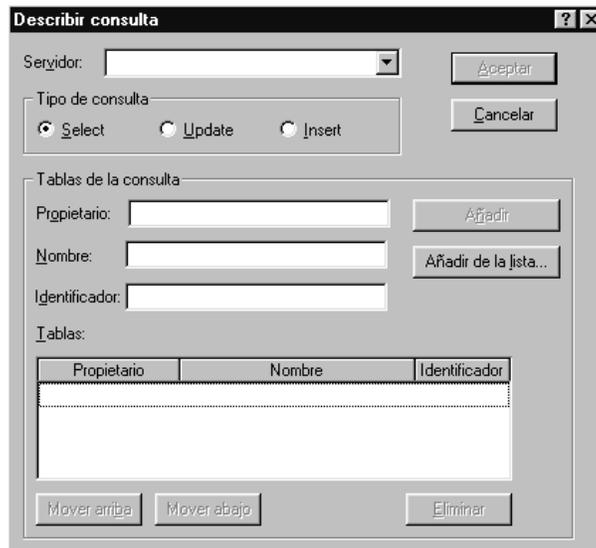
---

## Describir consultas

Utilice el mandato Describir consulta para crear documentos de consulta de SQL nuevos. Especifique uno o más nombres de tabla y el tipo de sentencia SQL que desea y QMF para Windows creará automáticamente una sentencia SQL que hace referencia a los nombres y tipos de datos de las columnas de la tabla.

## Creación de consultas de SQL nuevas

1. En el menú **Archivo**, pulse en **Describir consulta**. Se abrirá el recuadro de diálogo **Describir consulta**.



2. Seleccione el tipo de consulta que desea crear:

Tipo de consulta	Resultado
Select	Recuperar filas de una o más tablas.
Update	Cambiar la información de una tabla.
Insert	Añadir filas nuevas a una tabla.

3. Entre el propietario y el nombre de la tabla que se debe consultar.

**Nota:** Puede utilizar patrones para seleccionar nombres de tabla en una lista de tablas coincidentes.

- Utilice el carácter de porcentaje (%) para comparar una serie de cualquier longitud que contenga cualquier carácter. Por ejemplo, para listar todas las tablas cuyo nombre empieza con la letra A, entre A%.
- Utilice el carácter de subrayado (\_) para comparar un carácter individual. Por ejemplo, para listar todas las tablas cuyo propietario tiene la letra A en la segunda posición, entre \_A%.

Una vez que haya entrado un patrón, pulse en **Añadir de la lista** y seleccione una tabla de la lista resultante.

4. Entre un identificador exclusivo para la tabla.
5. Pulse en **Añadir**. La tabla se añadirá a la consulta.
6. Cuando haya añadido la tabla o las tablas que desea consultar, pulse en **Aceptar**. Se creará y se visualizará una consulta de SQL para las tablas seleccionadas.

---

## Variables de sustitución en consultas de SQL

Con las variables de sustitución, puede utilizar la misma consulta para recuperar información diferente proporcionando valores diferentes cada vez que ejecuta la consulta. Para recuperar un conjunto de datos distinto, no es necesario volver a escribir la consulta. En lugar de ello, simplemente facilite valores distintos para las variables de sustitución en la consulta al ejecutarla.

Una variable de sustitución es texto que se incluye en una consulta. Debe comenzar con un carácter & (&), y puede contener hasta 18 caracteres, los cuales pueden ser alfabéticos, numéricos o alguno de los caracteres especiales siguientes: ^ ! \$ % & ' { } ? @ # % \ o \_ . Por ejemplo, los valores siguientes son variables de sustitución válidas:

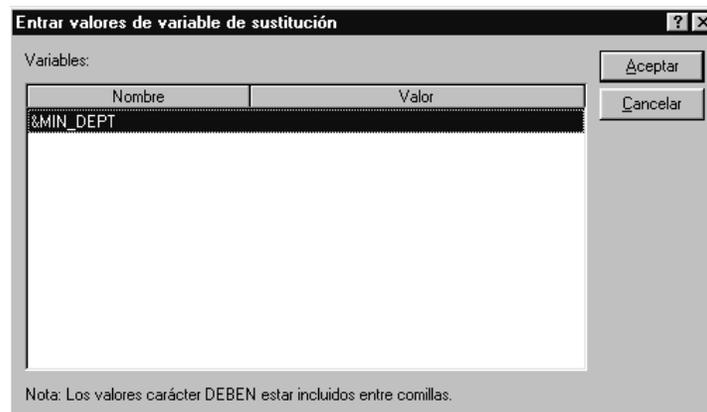
```
&VARIABLE1  
&NÚMERO_DEPARTAMENTO
```

Una variable de sustitución puede aparecer en cualquier lugar de una consulta y su valor puede ser cualquier elemento que se pueda escribir en una consulta (salvo un comentario). Por ejemplo, puede utilizar una variable de sustitución en lugar de un nombre de columna, una condición de búsqueda, una subconsulta o cualquier valor específico.

## Ejecución de consultas de SQL con variables de sustitución

1. Abra un documento de consulta nuevo y entre esta sentencia SQL:  

```
SELECT * FROM Q.PLANTILLA WHERE DTO >= &DTO_MIN
```
2. Ejecute la consulta. Se abrirá el recuadro de diálogo Entrar valores de variable de sustitución.



3. En el campo **Valor**, entre un valor de 50 y pulse en **Aceptar**. La consulta se ejecuta y visualiza los resultados de consulta.

Intente hacer pruebas con variables de sustitución reemplazando valores en las cláusulas SELECT y FROM. Compruebe qué resultados devuelven las consultas con entradas diferentes.

---

## Cómo guardar y abrir consultas de SQL

Las consultas pueden guardarse en el PC, en un servidor de archivos o en un servidor de bases de datos.

## Cómo guardar consultas de SQL en archivos

1. En una consulta abierta, pulse en el botón **Guardar** de la barra de herramientas.



Si ya se ha guardado previamente la consulta, se volverá a guardar. De lo contrario, se abrirá el recuadro de diálogo Guardar como.

2. Entre el nombre del archivo donde desea almacenar la consulta y pulse en **Aceptar**. Se guardará la consulta.

## Cómo abrir archivos de consultas de SQL guardados

1. Pulse en el botón **Abrir** de la barra de herramientas.



Se abrirá el recuadro de diálogo Abrir.

2. Seleccione el archivo que desea abrir y pulse el botón en **Aceptar**. La consulta seleccionada se abrirá en un documento de consulta de SQL nuevo.

## Cómo guardar consultas de SQL en el servidor de bases de datos

Las consultas que se guardan en el servidor pueden estar a disposición del resto de usuarios. Si desea compartir sus consultas con otros usuarios, guárdelas en el servidor de bases de datos.

1. En una consulta abierta, pulse en el botón **Guardar en servidor** de la barra de herramientas.



Se abrirá el recuadro de diálogo Guardar consulta.

El recuadro de diálogo 'Guardar Consulta' tiene un título con iconos de ayuda y cerrar. Incluye los siguientes campos y controles:

- Campo 'Propietario:' con un ícono de usuario y un campo de texto.
- Campo 'Nombre:' con un campo de texto.
- Campo 'Comentario:' con un área de texto grande.
- Botones 'Aceptar' y 'Cancelar'.
- Una casilla de verificación con el texto 'Compartir el objeto con otros usuarios'.

2. Entre el propietario y un nombre, seleccione si desea compartir o no la consulta con otros usuarios y pulse en **Aceptar**. La consulta se guardará en el servidor. Si ya existe una consulta con ese nombre, se le solicitará escribir encima de la consulta existente.

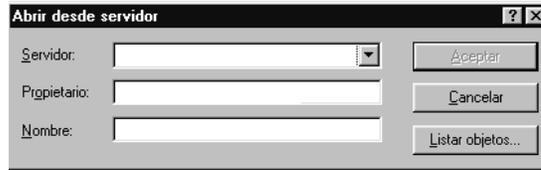
## Cómo abrir consultas de SQL guardadas en el servidor de bases de datos

Es posible abrir las consultas que haya guardado en el servidor de bases de datos.

1. Pulse en el botón **Abrir desde servidor** de la barra de herramientas.



Se abrirá el recuadro de diálogo Abrir desde servidor.



2. Entre un servidor, un propietario y un nombre, y pulse en **Aceptar**. Se abrirá la consulta de SQL.

---

## Impresión de consultas de SQL

Puede realizar una presentación preliminar de las consultas de SQL e imprimirlas.

### Presentación preliminar de una consulta

1. Abra una consulta y active la vista de SQL. Aparecerá la sentencia SQL.
2. En el menú **Archivo**, pulse en **Preparar página**. Se abrirá el recuadro de diálogo Preparar página.
3. Efectúe los cambios que desee en el diseño de la página y pulse en **Aceptar**.
4. Pulse en el botón **Presentación preliminar** en la barra de herramientas.



Aparecerá una presentación preliminar de la consulta impresa.

### Impresión de consultas de SQL

1. Abra una consulta y active la vista de SQL. Aparecerá la sentencia SQL.
2. En el menú **Archivo**, pulse en **Preparar página**. Se abrirá el recuadro de diálogo Preparar página.
3. Efectúe los cambios que desee en el diseño de la página y pulse en **Aceptar**.
4. Pulse en el botón **Imprimir** de la barra de herramientas.



Se imprimirá la consulta.

---

## Capítulo 3. Cómo trabajar con consultas solicitud

Las consultas solicitud constituyen un método sencillo para crear consultas seleccionando opciones en listas y menús. Una vez creada una consulta solicitud, el usuario puede guardarla o convertirla en una consulta de SQL.

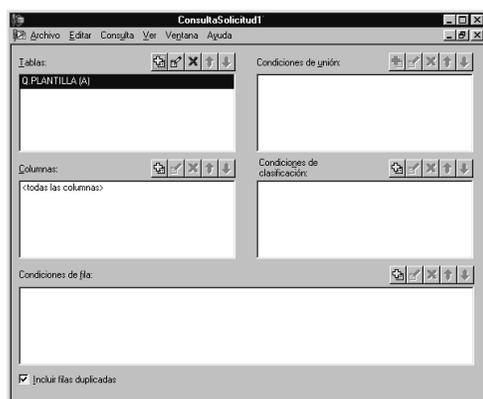
---

### Creación de consultas simples

La creación de consultas simples resulta una tarea fácil gracias a la interfaz para consultas solicitud.

#### Cómo abrir consultas solicitud nuevas

- En el menú **Archivo**, pulse en **Consulta solicitud nueva**. Se abrirá un nuevo documento de consulta solicitud.



#### Botones de acción de consulta solicitud

Las consultas solicitud se editan utilizando los botones de acción de consulta. Encima de la sección correspondiente, aparece un conjunto de botones que las controlan.

Botón de acción de consulta solicitud	Aspecto	Resultado
Añadir		Púlselo para añadir un elemento a la consulta solicitud.
Cambiar		Púlselo para editar el elemento resaltado de la consulta.
Suprimir		Púlselo para suprimir el elemento seleccionado.
Mover arriba y Mover abajo		Púselos para mover el elemento seleccionado hacia arriba y hacia abajo en la consulta solicitud.

## Adición de tablas a las consultas solicitud

1. En la sección Tablas del documento Consulta solicitud, pulse en el botón **Añadir**.



Se abrirá el recuadro de diálogo Tablas.



2. Escriba el propietario y el nombre de la tabla que desea añadir y pulse en **Añadir**. La tabla se añadirá a la consulta.

**Nota:** Puede utilizar patrones para seleccionar objetos de una lista de objetos coincidentes.

- Utilice el carácter de porcentaje (%) para comparar una serie de cualquier longitud que contenga cualquier carácter. Por ejemplo, para listar todas las tablas cuyo nombre empieza con la letra A, entre A%.
- Utilice el carácter de subrayado (\_) para comparar un carácter individual. Por ejemplo, para listar todas las tablas cuyo propietario tiene la letra A en la segunda posición, entre \_A%.

Una vez que haya entrado un patrón, pulse en **Añadir de la lista** y seleccione una tabla de la lista resultante.

3. Añada las condiciones de tablas adicionales a la consulta y pulse en **Cerrar**. Aparecerá el documento Consulta solicitud con las nuevas tablas listadas.

## Ejecución de consultas solicitud

Una consulta solicitud se ejecuta del mismo modo que una consulta de SQL. Pulse en el botón Ejecutar consulta de la barra de herramientas.



Se ejecutará la consulta solicitud.

---

## Creación de consultas complejas

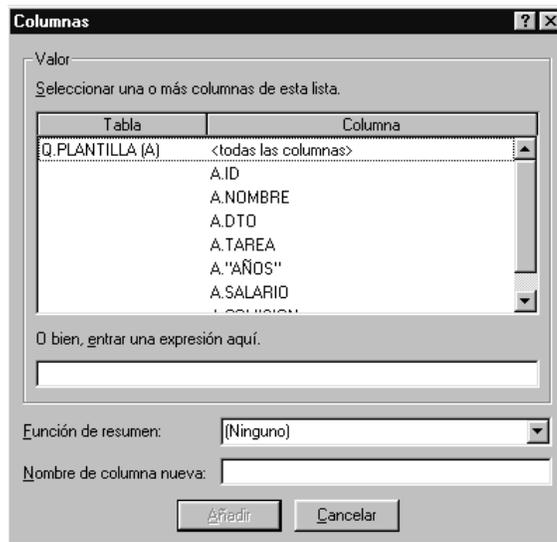
También es posible crear consultas más complejas gracias a la interfaz para consultas solicitud.

## Adición de columnas a consultas solicitud

1. En la sección Columna del documento Consulta solicitud, pulse en el botón **Añadir**.



Se abrirá el recuadro de diálogo Columnas.



2. Seleccione la columna que desea añadir y pulse en **Añadir**. La columna se añadirá a la consulta solicitud.
3. Añada las columnas adicionales a la consulta y pulse en **Cerrar**. Aparecerá el documento Consulta solicitud con las nuevas columnas listadas.

**Nota:** Puede aplicar una función de resumen a la columna seleccionando una en el campo Función. Las funciones de resumen disponibles son: AVG, COUNT, MAX, MIN y SUM.

**Nota:** Puede cambiar el nombre de una columna de la consulta escribiendo el nombre nuevo en el campo **Nombre de columna nueva**.

## Utilización de condiciones de clasificación

Las condiciones de clasificación se utilizan para especificar el modo en que se desea clasificar las filas de la consulta. Las filas pueden clasificarse por orden ascendente (A-Z) o descendente (Z-A).

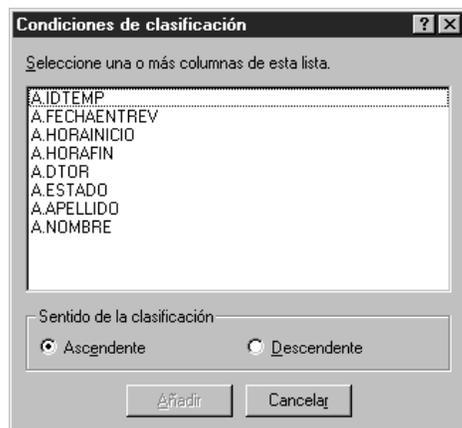
Si clasifica las filas por más de una columna, la primera columna se ordena primero, la segunda columna se ordena dentro del orden de la primera, y así sucesivamente.

## Adición de condiciones de clasificación

1. En la sección Condiciones de clasificación del documento Consulta solicitud, pulse en el botón **Añadir**.



Se abrirá el recuadro de diálogo Condiciones de clasificación.



2. Seleccione la columna según la cual desea efectuar la clasificación, la dirección en la que desea efectuar la clasificación y pulse en **Añadir**. La condición de clasificación se añadirá a la consulta solicitud.
3. Añada las condiciones de clasificación adicionales a la consulta y pulse en **Cerrar**. Aparecerá el documento Consulta solicitud con las nuevas condiciones de clasificación listadas.

## Utilización de condiciones de fila

En multitud de ocasiones, sólo se desean ver determinadas columnas de una tabla. Para seleccionar la visualización de filas específicas, añada condiciones de filas. Si no utiliza condiciones de fila, se visualizarán todas las filas de la tabla.

Éstas son todas las condiciones de fila disponibles:

- Igual a
- Menor que
- Menor que o igual a
- Mayor que
- Mayor que o igual a
- Entre
- Empieza por
- Termina por
- Contiene
- NULO

Las condiciones de fila las controlan los operadores siguientes:

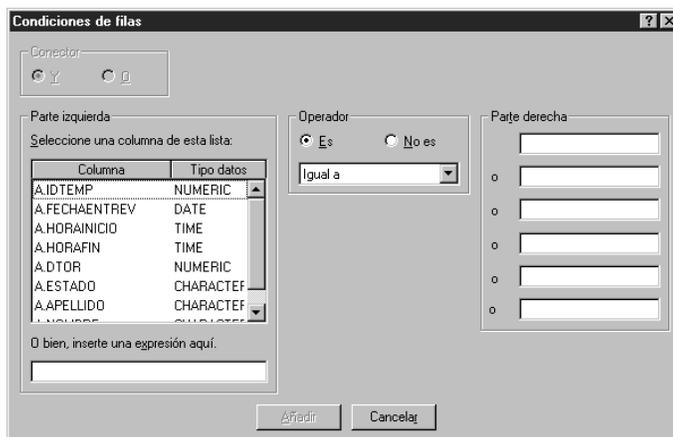
- Es
- No es

## Adición de condiciones de fila

1. En la sección Condiciones de fila del documento Consulta solicitud, pulse en el botón **Añadir**.



Se abrirá el recuadro de diálogo Condiciones de filas.



2. Seleccione las partes de la sentencia condicional y pulse en **Añadir**.

Parte de la condición de fila	Función
Parte izquierda	Seleccione la columna que desea examinar.
Operador	Determine la relación entre la parte izquierda y la parte derecha de la fila.
Parte derecha	Entre la condición que desea comprobar.

La condición de fila se añadirá a la consulta solicitud.

3. Añada las condiciones de fila adicionales a la consulta y pulse en **Cerrar**. Aparecerá el documento Consulta solicitud con las nuevas condiciones de fila listadas.

## Utilización de múltiples tablas en consultas solicitud

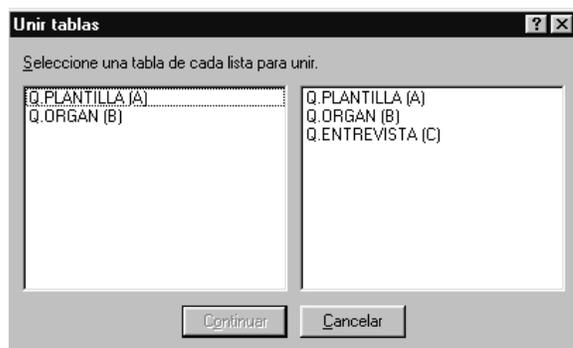
En una consulta solicitud es posible incluir información de más de una tabla.

Deberá relacionar las dos tablas, especificando una o más condiciones de unión en cada una de ellas. Sólo se incluyen en los resultados las filas de las tablas donde las condiciones de unión son iguales. El tipo de datos de cada columna de una condición de unión debe coincidir. Una vez que ha especificado la relación entre dos columnas, QMF para Windows recuerda la relación y la sugiere en posteriores consultas, haciendo que la tarea de creación de consultas subsiguientes resulte más simple y más eficiente.

## Creación de condiciones de unión de consulta solicitud

1. En la sección Tablas de la ventana Consulta solicitud, pulse en el botón **Añadir** para añadir un mínimo de dos tablas. Si las tablas no se han unido nunca, se abrirá el recuadro de diálogo Unir tablas. Si las tablas se han unido anteriormente, QMF para Windows sugiere la condición de unión que se haya

utilizado en esa ocasión.



2. Seleccione una columna con el mismo tipo de datos de cada tabla y pulse en **Añadir**. Aparecerá la condición de unión nueva en la consulta solicitud.

---

## Utilización de SQL en consultas solicitud

Puede utilizar la interfaz de consultas solicitud para aprender SQL.

### Visualización de SQL para consultas solicitud

En la vista de consulta solicitud, pulse en el botón **Ver SQL** de la barra de herramientas.



Aparecerá la sentencia SQL equivalente de la consulta solicitud. No puede modificar la sentencia SQL desde esta vista.

### Conversión de consultas solicitud en SQL

Una consulta solicitud puede convertirse en un documento de consulta de SQL nuevo. La consulta de SQL nueva se puede modificar, guardar, imprimir y ejecutar. En el menú **Consulta**, pulse en **Convertir a SQL**. La consulta se convierte en un nuevo documento de consulta de SQL.

---

## Utilización de variables de sustitución en consultas solicitud

Las variables de sustitución se pueden utilizar en una consulta solicitud del mismo modo que se utilizan en una consulta de SQL. Consulte el apartado "Variables de sustitución en consultas de SQL".

Por ejemplo, se pueden utilizar variables de sustitución en:

- una condición de fila  
DT0 es Mayor que o Igual a &DtoMin
- una especificación de columna  
&NumEntrada

---

## Cómo guardar consultas solicitud

Las consultas solicitud pueden guardarse en el PC, en un servidor de archivos o en un servidor de bases de datos.

## Cómo guardar consultas solicitud en archivos

1. En una consulta solicitud abierta, pulse en el botón **Guardar** de la barra de herramientas.



**Nota:** Si ya se ha guardado previamente la consulta, se volverá a guardar. De lo contrario, se abrirá el recuadro de diálogo Guardar como.

2. Entre el nombre del archivo donde desea almacenar la consulta solicitud y pulse en **Aceptar**. Se guardará la consulta.

## Cómo abrir archivos de consultas solicitud guardados

1. Pulse en el botón **Abrir** de la barra de herramientas.



Se abrirá el recuadro de diálogo Abrir.

2. Seleccione el archivo que desea abrir y pulse en **Aceptar**. La consulta solicitud seleccionada se abrirá en un documento de consulta de SQL nuevo.

## Cómo guardar consultas solicitud en el servidor de bases de datos

1. En una consulta solicitud abierta, pulse en el botón **Guardar en servidor** de la barra de herramientas.



Se abrirá el recuadro de diálogo Guardar consulta.

El recuadro de diálogo "Guardar Consulta" tiene un título "Guardar Consulta" y botones de ayuda y cerrar. Incluye los siguientes campos:

- Propietario: un campo de texto con el símbolo de propiedad.
- Nombre: un campo de texto.
- Comentario: un área de texto con flechas de desplazamiento.
- Una casilla de verificación "Compartir el objeto con otros usuarios" que está marcada.
- Botones "Aceptar" y "Cancelar" a la derecha.

2. Entre el propietario y un nombre, seleccione si desea compartir o no la consulta con otros usuarios y pulse en **Aceptar**. La consulta se guardará en el servidor. Si ya existe una consulta con ese nombre, se le solicitará grabar encima de la consulta existente.

## Cómo abrir consultas solicitud guardadas en el servidor de bases de datos

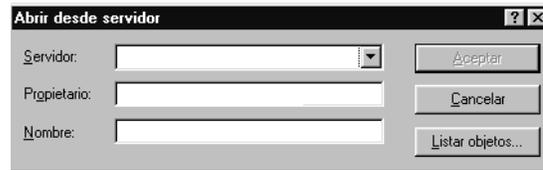
Es posible abrir las consultas solicitud que haya guardado en el servidor de bases de datos.

1. Pulse en el botón **Abrir desde servidor** de la barra de herramientas.



>

Se abrirá el recuadro de diálogo Abrir desde servidor.



2. Entre un servidor, un propietario y un nombre, y pulse en **Aceptar**. Se abrirá la consulta solicitud.

---

## Impresión de consultas solicitud

Puede imprimir la consulta solicitud. También puede imprimir el texto SQL de una consulta solicitud. Consulte el apartado “Impresión de consultas de SQL” en la página 12.

## Presentación preliminar de consultas solicitud

Puede obtener la presentación preliminar de los resultados o del texto de una consulta solicitud antes de imprimirla.

1. Abra una consulta y active la vista de consulta solicitud. Aparecerá la consulta.
2. En el menú **Archivo**, pulse en **Preparar página**. Se abrirá el recuadro de diálogo Preparar página.
3. Efectúe los cambios que desee en el diseño de la página y pulse en **Aceptar**.
4. Pulse en el botón **Presentación preliminar** en la barra de herramientas.



Aparecerá una presentación preliminar de la consulta impresa.

---

## Capítulo 4. Cómo trabajar con resultados de consultas

Se puede aplicar formato, agrupar y agregar directamente a los resultados de la consulta. Este formato se puede guardar con la consulta o ser exportado como un formato.

---

### Clasificación y ajuste de los resultados de una consulta

Los usuarios pueden seleccionar, reordenar y clasificar los resultados de los datos de una consulta, y ajustar el tamaño de los mismos.

#### Selección de columnas y filas

Una vez ejecutada la consulta, puede utilizar los controles de la vista de resultados para editar y seleccionar información.

Selectores de columna y fila	Aspecto	Función
Selector de fila		Pulse el botón para seleccionar todos los datos de una fila.
Selector de columna		Pulse el botón para seleccionar todos los datos de una columna.
Casilla		Pulse el botón directamente en la casilla para seleccionarla.
Botones Desplazar al final y Desplazar al principio		Pulse el botón para desplazarse al final o al principio de un conjunto de resultados de consulta.

#### Cambio de tamaño de las columnas y filas

Es posible cambiar el aspecto de un grupo de resultados de consulta cambiando el tamaño de sus columnas y filas.

1. Mediante el ratón, seleccione la línea divisoria en color negro situada entre dos columnas o dos filas.
2. Arrastre la línea divisoria de un lado al otro o de arriba a abajo para cambiar el tamaño de la columna o de la fila.

**Nota:** Si guarda la consulta después de cambiar el tamaño de las columnas o las filas, esta consulta siempre se visualizará con el nuevo formato.

#### Ajuste automático de columnas y filas

Puede cambiar automáticamente el tamaño de las columnas y de las filas para que se ajusten a los datos que contienen.

Mediante el ratón, seleccione toda una columna o fila y efectúe una doble pulsación en la línea divisoria en color negro que la separa del objeto contiguo. El tamaño de la columna se ajustará automáticamente a los datos que contiene.

**Nota:** Si guarda la consulta después de cambiar el tamaño de las columnas o las filas, esta consulta siempre se visualizará con el nuevo formato.

## Clasificación de resultados de consulta

Una vez ejecutada la consulta, puede clasificar los resultados por columna alfabéticamente.

En la vista de Resultados de una consulta, seleccione una columna y seleccione **Clasificación ascendente** en el menú **Resultados**.

Los resultados de consulta se clasifican por orden ascendente.

-o bien-

En la vista de Resultados de una consulta, seleccione una columna y seleccione **Clasificación descendente** en el menú **Resultados**.

Los resultados de consulta se clasifican por orden descendente.

**Nota:** Para aplicar una clasificación más compleja a la columna seleccionada, seleccione **Clasificar** en el menú **Resultados**.

## Reordenación de columnas

Puede cambiar el orden de las columnas en los resultados de consulta.

En la vista Resultados de una consulta, seleccione una columna y arrástrela hasta su nueva ubicación.

La columna aparecerá en el nuevo orden.

---

## Cómo aplicar formato a resultados de consulta

Es posible cambiar la fuente utilizada para visualizar consultas y resultados de consulta. La selección de fuentes dependerá de lo que se haya instalado en su sistema. Para obtener más información sobre cómo añadir fuentes, consulte el recurso de ayuda del sistema operativo.

**Nota:** Si guarda la consulta después de seleccionar una fuente nueva para visualizar los resultados de consulta, estos resultados siempre se visualizarán con la nueva fuente.

## Selección de la fuente de visualización de resultados de consulta

1. En la vista de Resultados, seleccione **Establecer fuente** en el menú **Resultados**. Se abrirá el recuadro de diálogo Fuente.
2. Seleccione la fuente y el tamaño de tipo para visualizar los resultados de la consulta y pulse en **Aceptar**. Los resultados de consulta se visualizan en el formato que ha especificado.

**Nota:** Pulse en **Establecer como valor por omisión** para utilizar la fuente seleccionada como fuente por omisión para todos los resultados de consulta.

## Cómo aplicar formato a resultados de consulta numéricos

1. En la vista de Resultados, seleccione una columna que contenga valores numéricos y seleccione **Formato** en el menú **Resultados**. Se abrirá el recuadro de diálogo Formato.
2. Especifique el formato que desee aplicar y pulse en **Aceptar**. Se dará formato a los valores según la selección realizada.

**Nota:** Pulse en **Establecer como valor por omisión** para utilizar la fuente seleccionada como fuente por omisión para todos los resultados de consulta.

## Conversión de los valores de formato de los resultados de consulta en un formato

Puede convertir los valores de formato de los resultados de consulta en un formato.

1. En el menú **Resultados**, seleccione **Visualizar informe**.  
Se abrirá el recuadro de diálogo Seleccionar formato.
2. Seleccione **Desde consulta** y pulse en **Aceptar**.  
Los valores de formato de los resultados de consulta se convertirá en un formato y se abrirán en una nueva ventana de formato.

---

## Cómo agrupar y agregar resultados de consulta

Puede agrupar y agregar los resultados de consulta y aplicarles formato de resumen.

### Agrupación de resultados de consulta

Puede agrupar los resultados de consulta con o sin información de resumen.

1. Seleccione la columna que desee agrupar.
2. En el menú **Resultados**, seleccione el tipo de agrupación que desee aplicar.  
La columna se agrupará según la selección realizada.

### Cómo resumir resultados de consulta

Puede resumir los resultados de consulta por columna.

1. Seleccione la columna que desee agrupar.
2. En el menú **Resultados**, seleccione el tipo de resumen que desee aplicar.  
La columna se resumirá según la selección realizada.

---

## Cómo guardar resultados de consulta y el formato

Puede guardar resultados de consulta y guardar los valores de formato como un formato.

### Cómo guardar resultados de consulta como una tabla

Puede guardar los resultados de consulta como una tabla en un servidor de bases de datos.

1. En el menú **Resultados**, seleccione **Guardar en base de datos**.  
Se abrirá el recuadro de diálogo Guardar datos.
2. Entre un propietario y un nombre de tabla y pulse en **Aceptar**.

Los resultados de consulta se guardarán como una tabla en la base de datos.

## Cómo guardar resultados de consulta en archivos

Puede guardar los resultados de consulta en un archivo en su PC o en un servidor de archivos.

1. En el menú **Resultados**, seleccione **Guardar en archivo**.  
Se abrirá el recuadro de diálogo Exportar datos.
2. Especifique la ubicación en la cual desee guardar el archivo y las opciones de exportación, y pulse **Aceptar**.  
Los resultados de consulta se guardarán en un archivo.

---

## Impresión de resultados de consulta

Puede realizar una presentación preliminar de los resultados de consulta e imprimirlos.

### Presentación preliminar de resultados de consulta

1. Abra y ejecute una consulta. Aparecerán los resultados de la consulta.
2. En el menú **Archivo**, seleccione **Preparar página**. Se abrirá el recuadro de diálogo Preparar página.
3. Efectúe los cambios que desee en el diseño de la página y pulse en **Aceptar**.
4. Pulse en el botón **Presentación preliminar** en la barra de herramientas.



Aparecerá una presentación preliminar de los resultados de consulta impresos.

### Impresión de resultados de consulta

1. Abra una consulta y active la vista de Resultados. Aparecerán los resultados de la consulta.
2. En el menú **Archivo**, seleccione **Preparar página**. Se abrirá el recuadro de diálogo Preparar página.
3. Efectúe los cambios que desee en el diseño de la página y pulse en **Aceptar**.
4. Pulse en el botón **Imprimir** de la barra de herramientas.



Se imprimirán los resultados de la consulta.

---

## Capítulo 5. Cómo trabajar con informes

Los informes se crean al combinar resultados de consulta con los valores de formato de un formato.

---

### Formatos

Los formatos son una serie de instrucciones de formato que se utilizan para crear, visualizar e imprimir informes.

#### Interpretación de los formatos

Los formatos constan de diversos componentes. Todos estos componentes pueden editarse en un documento de formato.

##### Principal

Componentes primarios de un formato, incluidos los encabezamientos, pies de página y cortes.

**Cortes** Características, contenido y colocación de un máximo de seis líneas de subtotales en un informe.

##### Cálculos

Expresiones de cálculo de informe.

**Nota:** Para poder utilizar cálculos de formato, debe tener instalado ObjectREXX de IBM en su máquina.

##### Columnas

Aspecto y formato de las columnas del informe. Las características que se pueden definir incluyen el orden, la forma, el uso, el sangrado y el ancho de columna.

##### Condiciones

Restricciones de formato condicionales. Por ejemplo, puede definir el formato para que no se visualicen filas que no cumplan determinadas características.

##### Detalles

Encabezamientos de detalle y texto de bloque de informe. Es aquí donde puede combinar datos de tablas con texto de formato libre o sustituir dichos datos por este tipo de texto para crear cartas de formato o etiquetas de direcciones.

**Final** Contenido y colocación del texto final del informe. Por ejemplo, puede elegir incluir texto final y datos de resumen al final del informe.

##### HTML

Contenido y colocación de formato y códigos HTML en informes HTML.

##### Opciones

Opciones de aspecto diversas del informe.

##### Página

Contenido y colocación del encabezamiento y el pie de página del informe.

## Cómo producir un informe utilizando un formato

Los informes se crean combinando los resultados de consulta con las opciones de formato que se incluyen en un formato. Es posible producir múltiples informes a partir de un único conjunto de resultados de consulta mediante la repetición de este proceso.

1. Desde una vista de resultados de consulta, pulse en el botón **Visualizar informe**.



Se abrirá el recuadro de diálogo Seleccionar formato.



2. Dependiendo del tipo de formato que haya seleccionado en el recuadro de diálogo Seleccionar formato, se le solicitará que facilite información adicional. Especifique la ubicación del archivo, el propietario y el nombre, o bien el título del documento, según convenga, y pulse en **Aceptar**. El informe se generará utilizando el formato seleccionado y los resultados de consulta actuales.

## Edición de un formato

La ventana Formato proporciona muchas opciones para editar y cambiar el aspecto de los formatos.

Desde un formato abierto, visualice el menú Formato. Este menú contiene todas las opciones que puede utilizar para editar y cambiar el formato del formato. También puede editar cualquiera de estos componentes pulsando en el botón correspondiente en la barra de herramientas.

---

## Creación de un formato

Todos estos pasos incluyen datos de ejemplos de la tabla Q.PLANTILLA. Haga pruebas con valores diferentes para crear sus propios formatos personalizados.

### Paso 1: Crear un formato

1. Ejecute la consulta de SQL siguiente para recuperar los datos para visualizar en el informe:

```
SELECT * FROM Q.PLANTILLA ORDER BY DTO, NOMBRE
```

Aparecerán los resultados de la consulta.

2. Pulse en el botón **Visualizar informe** de la barra de herramientas. Se abrirá el recuadro de diálogo Seleccionar formato.
3. Especifique que desea utilizar el formato por omisión y pulse en **Aceptar**. QMF para Windows visualizará el informe por omisión. Para efectuar cambios en el formato por omisión, pulse en alguno de los botones de componentes del formato de la barra de herramientas. En la barra de herramientas del formato aparece un botón para cada componente.

## Paso 2: Cambiar el orden de las columnas

Deseamos que NOMBRE sea la primera columna del informe y que ID sea la segunda. El orden de las columnas se especifica en el componente Columnas del formato.

1. Pulse en **Columnas...** en el menú **Formato** para visualizar la pestaña Columnas en el recuadro de diálogo Formato.
2. Cambie la secuencia de una columna escribiendo encima del valor de secuencia existente. Para que NOMBRE sea la primera columna del informe, cambie el número de secuencia (la columna de la lista etiquetada Sec) por 1.
3. Para que ID sea la segunda columna del informe, cambie el número de secuencia por 2 y pulse en **Aceptar**. QMF para Windows visualizará el informe con el orden de columnas nuevo en la ventana Formato.

## Paso 3: Cambiar los encabezamientos de columna

Deseamos que EMPLEADO sea el primer encabezamiento de columna y COMISION sea el segundo. El texto de encabezamiento de columna se especifica en el componente Columnas del formato.

1. Pulse en **Columnas...** en el menú **Formato** para visualizar la pestaña Columnas en el recuadro de diálogo Formatos.
2. Cambie el encabezamiento de columna escribiendo encima del texto de encabezamiento de columna existente. Cambie el primer encabezamiento de columna por EMPLEADO y el último por COMISION y pulse en **Aceptar**. QMF para Windows visualizará el informe con los encabezamientos de columna nuevos en la ventana Formato.

## Paso 4: Cambiar el formato de columna

Deseamos que la columna SALARIO se visualice con el símbolo de moneda apropiado. El formato de una columna está determinado por su código de edición, el cual se especifica en el componente Columnas del formato.

1. Pulse en **Columnas...** en el menú **Formato** para visualizar la pestaña Columnas en el recuadro de diálogo Formatos.
2. Cambie el código de edición de la columna SALARIO por D2 escribiendo encima del código de edición existente y pulse en **Aceptar**. QMF para Windows visualizará el informe utilizando el símbolo de moneda apropiado en la columna SALARIO en la ventana Formato.

## Paso 5: Añadir información de resumen

Deseamos dividir el informe en secciones por separado, una para cada departamento. Además, queremos ver el total de SALARIO y COMISION de cada departamento al final de cada sección. Para ello, necesitamos especificar cómo deseamos que se utilice cada columna del informe. El uso de una columna está determinado por su código de uso, el cual se especifica en el componente Columnas del formato.

1. Pulse en **Columnas...** en el menú **Formato** para visualizar la pestaña Columnas en el recuadro de diálogo Formatos.
2. Para dividir el informe en secciones basadas en DTO, cambie el código de uso DTO por CORTE1. Los códigos de uso que empiezan por la palabra CORTE producen un corte de sección de la columna especificada. El número que sigue a la palabra CORTE determina el nivel de corte; en un informe se soporta un máximo de seis niveles de corte.

3. Para especificar que deseamos incluir un total de SALARIO y COMISION para cada DTO, cambie el código de uso de SALARIO y de COMISION por SUMA.
4. El informe resultará más fácil de interpretar si también incluimos información descriptiva al final de cada corte de sección. Para ello, pulse en **Cortes...** en el menú **Formato**.
5. El texto de pie de corte se especifica en la pestaña Cortes del recuadro de diálogo Formato. Establezca Total de departamento como la primera línea de pie de corte y pulse en **Aceptar**. QMF para Windows visualizará la ventana Formato.

## Paso 6: Añadir encabezamientos y pies de página

Deseamos añadir un encabezamiento y un pie de página a nuestro informe. Los encabezamientos y los pies de página se especifican en el componente Página del formato.

1. Pulse en **Página...** en el menú **Formato** para visualizar la pestaña Página en el recuadro de diálogo Formato.
2. La parte superior de este diálogo se utiliza para especificar el encabezamiento de página. Establezca Informe de departamento como la primera línea del encabezamiento de página y Total de salario y comisiones como la segunda línea. Elija cómo desea que se alinee el encabezamiento.
3. La parte inferior de este diálogo se utiliza para especificar el pie de página. Establezca Fin de página como la primera línea del pie de página. Elija cómo desea que se alinee el pie y pulse en **Aceptar**. QMF para Windows visualizará la ventana Formato.

---

## Cómo guardar formatos

Los formatos pueden guardarse en el PC, en un servidor de archivos o en un servidor de bases de datos.

### Cómo guardar un formato en un archivo

1. En un formato abierto, pulse en el botón **Guardar**.
2. Si el formato se ha guardado anteriormente, seleccione **Guardar**. De lo contrario, se abrirá el recuadro de diálogo Guardar como.
3. Entre el nombre del archivo donde desea almacenar el formato y pulse en **Aceptar**. Se guardará el formato.

### Cómo abrir un archivo de formato guardado

1. Pulse en el botón **Abrir** de la barra de herramientas.



Se abrirá el recuadro de diálogo Abrir.

2. Seleccione el archivo que desea abrir y pulse en **Aceptar**. El formato seleccionado se abrirá en un documento de formato nuevo.

### Cómo guardar formatos en el servidor de bases de datos

Los formatos que se guardan en el servidor pueden estar a disposición del resto de los usuarios. Si desea compartir sus formatos con otros usuarios, guárdelos en el servidor de bases de datos.

1. En un formato abierto, pulse en el botón **Guardar en servidor** de la barra de herramientas.



Se abrirá el recuadro de diálogo Guardar formato.

El diálogo 'Guardar Consulta' tiene un título con iconos de ayuda y cerrar. Incluye los siguientes elementos:

- Campo 'Propietario:' con un ícono de clave y un botón 'Aceptar'.
- Campo 'Nombre:' con un botón 'Cancelar'.
- Campo 'Comentario:' con un área de texto y flechas de desplazamiento.
- Una casilla de verificación 'Compartir el objeto con otros usuarios' que está marcada.

2. Entre un propietario, un nombre, seleccione si desea compartir o no el formato con otros usuarios y pulse en **Aceptar**. El formato se guardará en el servidor. Si ya existe un formato con ese nombre, se le solicitará grabar encima del formato existente.

## Cómo abrir formatos guardados en el servidor de bases de datos

Es posible abrir los formatos que haya guardado en el servidor de bases de datos.

1. Pulse en el botón **Abrir desde servidor** de la barra de herramientas.



Se abrirá el recuadro de diálogo Abrir desde servidor.

El diálogo 'Abrir desde servidor' tiene un título con iconos de ayuda y cerrar. Incluye los siguientes elementos:

- Campo 'Servidor:' con un menú desplegable y un botón 'Aceptar'.
- Campo 'Propietario:' con un botón 'Cancelar'.
- Campo 'Nombre:' con un botón 'Listar objetos...'.

2. Entre un servidor, un propietario y un nombre, y pulse en **Aceptar**. Se abrirá el formato.

---

## Impresión de informes

Los informes producidos pueden imprimirse.

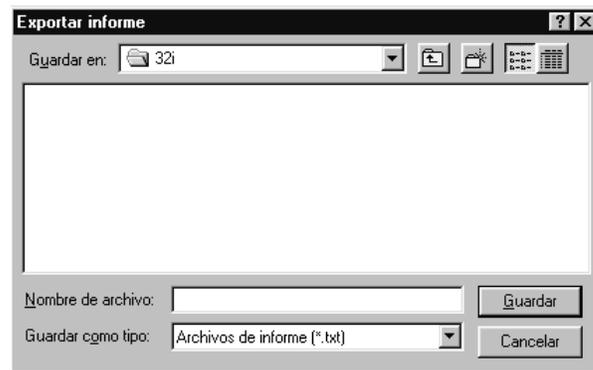
1. Abra un formato y pulse en **Preparar página**.
2. Efectúe los cambios que desee en el diseño de la página y pulse en **Aceptar**.
3. Pulse en **Imprimir informe** en el menú **Archivo**. Se imprimirá el informe.

---

## Exportación de informes

El informe se puede exportar a un archivo.

1. Abra un formato y pulse el botón en **Preparar página**.
2. Efectúe los cambios que desee en el diseño de la página y pulse en **Aceptar**.
3. Pulse en **Exportar informe** en el menú **Archivo**. Se abrirá el recuadro de diálogo Exportar informe.



4. Entre el nombre del archivo donde desea almacenar el informe y pulse en **Aceptar**. El informe se exportará.

---

## Capítulo 6. Cómo trabajar con procedimientos

Los procedimientos lineales le permiten ejecutar consultas, generar informes, editar datos y llevar a cabo otras funciones mediante la ejecución de un solo mandato. Si desea obtener una lista completa de todos los mandatos de procedimiento que soporta QMF para Windows, consulte el recurso de ayuda en línea.

Los procedimientos con lógica, o procedimientos REXX, son similares a los procedimientos lineales, pero contienen lenguaje de programación Object REXX de IBM así como mandatos de procedimiento. Object REXX debe estar instalado localmente para poder ejecutar procedimientos con lógica.

---

### Ejecución de procedimientos

Los procedimientos se utilizan para ejecutar múltiples funciones con un mandato.

#### Creación de un procedimiento lineal nuevo

En el menú **Archivo**, seleccione **Nuevo procedimiento**.

Se abrirá un nuevo procedimiento.

#### Creación de un procedimiento con lógica nuevo

1. En el menú **Archivo**, seleccione **Nuevo procedimiento**.  
Se abrirá un nuevo procedimiento.
2. Escriba una línea de comentario REXX como la primera línea del procedimiento. Las líneas de comentario REXX empiezan con `/*` y terminan con `*/`.
3. Escriba los mandatos de procedimiento QMF que desee en el procedimiento. Los mandatos de QMF se deben entrar en mayúsculas y entre comillas.
4. Escriba los mandatos REXX que desee en el procedimiento.

**Nota:** Los mandatos REXX se ejecutan localmente y no en un servidor de bases de datos. Object REXX debe estar instalado localmente.

#### Ejecución de un procedimiento en un servidor de bases de datos

1. Abra un documento de procedimiento nuevo y escriba un conjunto de mandatos, o bien abra un procedimiento existente en un archivo o en el servidor de bases de datos.
2. Pulse en el botón **Ejecutar procedimiento** de la barra de herramientas.



Se ejecutará el procedimiento.

---

## Cómo guardar procedimientos

Los procedimientos pueden guardarse en el PC, en un servidor de archivos o en un servidor de bases de datos.

### Cómo guardar procedimientos en archivos

1. En un procedimiento abierto, pulse en el botón **Guardar** de la barra de herramientas.



Si el procedimiento se ha guardado anteriormente, se guardará ahora. De lo contrario, se abrirá el recuadro de diálogo Guardar como.

2. Entre el nombre del archivo donde desea almacenar el procedimiento y pulse en **Aceptar**. Se guardará el procedimiento.

### Cómo abrir un archivo de procedimiento guardado

1. Pulse en el botón **Abrir** de la barra de herramientas.



Aparecerá el recuadro de diálogo Abrir.

2. Seleccione el archivo que desea abrir y pulse en **Aceptar**. El procedimiento seleccionado se abrirá en un documento de procedimiento nuevo.

### Cómo guardar un procedimiento en el servidor de bases de datos

1. En un procedimiento abierto, pulse en el botón **Guardar en servidor** de la barra de herramientas.



Se abrirá el recuadro de diálogo Guardar Consulta.

El recuadro de diálogo "Guardar Consulta" tiene un título con iconos de ayuda y cerrar. Incluye los siguientes campos y controles:

- Campo "Propietario:" con un ícono de usuario y un campo de texto.
- Campo "Nombre:" con un campo de texto.
- Campo "Comentario:" con un área de texto grande.
- Botones "Aceptar" y "Cancelar" a la derecha.
- Una casilla de verificación "Compartir el objeto con otros usuarios" marcada.

2. Entre un propietario, un nombre, seleccione si desea compartir o no el procedimiento guardado con otros usuarios y pulse en **Aceptar**. El procedimiento se guardará en el servidor.

Si ya existe un procedimiento con ese nombre, se le solicitará grabar encima del procedimiento existente.

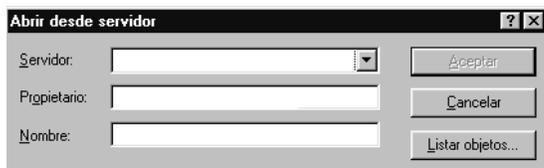
## Cómo abrir procedimientos guardados en el servidor de bases de datos

Es posible abrir los procedimientos que haya guardado en el servidor de bases de datos.

1. Pulse en el botón **Abrir desde servidor** de la barra de herramientas.



Se abrirá el recuadro de diálogo **Abrir desde servidor**.



2. Entre un servidor, un propietario y un nombre y pulse en **Aceptar**. Se abrirá el procedimiento.

---

## Impresión de procedimientos

El texto de un procedimiento puede imprimirse.

### Presentación preliminar de un procedimiento

1. Abra un procedimiento. Aparecerán los mandatos de procedimiento.
2. En el menú **Archivo**, pulse el botón en **Preparar página**. Se abrirá el recuadro de diálogo **Preparar página**.
3. Efectúe los cambios que desee en el diseño de la página y pulse en **Aceptar**.
4. Pulse en el botón **Presentación preliminar** de la barra de herramientas:



Aparecerá una presentación preliminar del procedimiento impreso.

### Impresión de un procedimiento

1. Abra un procedimiento. Aparecerán los mandatos de procedimiento.
2. En el menú **Archivo**, pulse en **Preparar página**. Se abrirá el recuadro de diálogo **Preparar página**.
3. Efectúe los cambios que desee en el diseño de la página y pulse en **Aceptar**.
4. Pulse en el botón **Imprimir** de la barra de herramientas:



Se imprimirá el procedimiento.



---

## Capítulo 7. Cómo trabajar con listas

Las listas le ofrecen un modo sencillo de ver grupos de objetos de QMF.

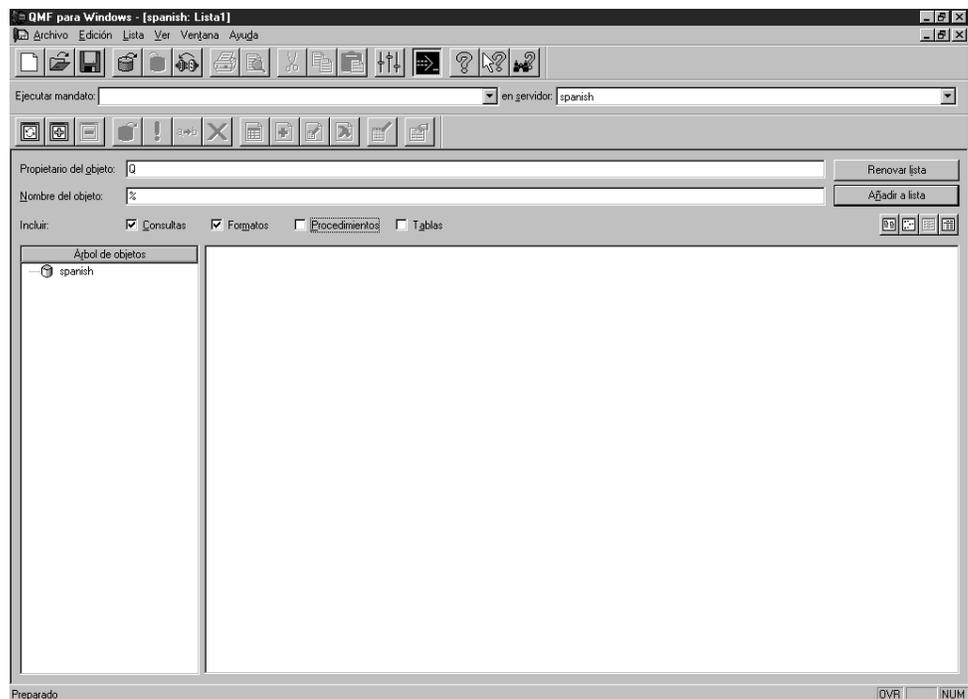
---

### Objetos

QMF para Windows reconoce cuatro tipos de objetos: consultas, formatos, procedimientos y tablas. Puede utilizar la ventana Lista para ver los objetos basándose en el nombre del objeto, el propietario del objeto y el tipo de objeto.

### Listado de objetos

1. En el menú **Archivo**, seleccione **Nueva Lista**.  
Se abrirá la ventana Lista.



2. Especifique un propietario y un nombre.

**Nota:** Puede utilizar patrones para seleccionar objetos de una lista de objetos coincidentes.

- Utilice el carácter de porcentaje (%) para comparar una serie de cualquier longitud que contenga cualquier carácter. Por ejemplo, para listar todas las tablas cuyo nombre empieza con la letra A, entre A%.
  - Utilice el carácter de subrayado (\_) para comparar un carácter individual. Por ejemplo, para listar todas las tablas cuyo propietario tiene la letra A en la segunda posición, entre \_A%.
3. Seleccione el tipo de objeto que está buscando.
  4. Pulse el botón en **Renovar lista**. Se visualizará una lista de objetos guardados en el servidor de bases de datos y que cumplen las condiciones.

---

## Mandatos de la ventana Lista

Al pulsar el botón derecho en un objeto de la ventana Lista se visualiza una lista de mandatos idénticos a los que figuran en el menú Lista.

### Visualizar objeto

Abre el objeto seleccionado para visualizarlo. Disponible para consultas, formatos, procedimientos y tablas.

### Ejecutar objeto

Ejecuta el objeto seleccionado. Disponible para consultas y procedimientos.

### Describir objeto

Crea una consulta basada en la tabla seleccionada. Puede optar por describir una consulta SELECT de SQL, una consulta UPDATE de SQL, una consulta INSERT de SQL o una consulta solicitud. Disponible para tablas.

### Editar objeto

Abre el objeto seleccionado para editarlo. Disponible para tablas.

### Propiedades

Visualiza las propiedades del objeto seleccionado, entre las que se incluyen comentarios, atributos e información de uso histórica. Disponible para consultas, formatos, procedimientos y tablas.

---

## Creación de listas

Las listas se pueden utilizar como grupos de objetos. Por ejemplo, cree una lista de todas las consultas, formatos, procedimientos y tablas relacionados con el inventario para poder mantener todo el trabajo en un solo lugar. Una vez creada, puede añadir y eliminar objetos de la lista y guardarla para utilizarla posteriormente.

### Cómo añadir objetos a las listas

Puede añadir objetos a listas.

En una lista abierta, especifique la información del nombre y del propietario de los objetos que desee añadir y pulse en el botón **Añadir a lista** en la barra de herramientas.



Los objetos que se correspondan con el propietario y el nombre especificados se añadirán a la lista.

### Eliminación de objetos de las listas

Es posible eliminar los objetos no relacionados de las listas.

En una lista abierta, pulse en el botón **Eliminar** de la barra de herramientas.



El objeto se eliminará de la lista, pero no se suprimirá.

## Guardar listas en archivos

1. En una lista abierta, pulse en el botón **Guardar** de la barra de herramientas.



Si la lista no se ha guardado anteriormente, se guardará ahora. De lo contrario, se abrirá el recuadro de diálogo Guardar como.

2. Entre el nombre del archivo donde desea almacenar la lista y pulse en **Aceptar**. La lista se guardará.

## Cómo abrir archivos de lista guardados

1. Pulse en el botón **Abrir** de la barra de herramientas.



Se abrirá el recuadro de diálogo Abrir.

2. Seleccione el archivo que desea abrir y pulse en **Aceptar**. Se abrirá la lista seleccionada en un documento de lista.



---

## Capítulo 8. Cómo trabajar con archivos de trabajos

Puede planificar y ejecutar procedimientos utilizando archivos de trabajos. Los archivos de trabajos utilizan el planificador de Windows para ejecutar procedimientos según la fecha y la hora preestablecidas.

---

### Archivos de trabajos

Puede crear archivos de trabajo y almacenarlos localmente o en el servidor de bases de datos.

#### Creación de archivos de trabajos

1. En el menú **Archivo**, seleccione **Nuevo trabajo**.  
Se abrirá un nuevo documento de trabajo.

#### Ejecución de archivos de trabajos

Puede ejecutar archivos de trabajos que se han guardado localmente.

1. Abra el archivo de trabajos.
2. Pulse en el botón **Ejecutar trabajo** en la barra de tareas.



3. Arrastre la línea divisoria de un lado al otro o de arriba a abajo para cambiar el tamaño de la columna o de la fila.

**Nota:** Si guarda la consulta después de cambiar el tamaño de las columnas o las filas, esta consulta siempre se visualizará con el nuevo formato.

#### Ajuste automático de columnas y filas

Puede cambiar automáticamente el tamaño de las columnas y de las filas para que se ajusten a los datos que contienen.

Mediante el ratón, seleccione toda una columna o fila y efectúe una doble pulsación en la línea divisoria en color negro que la separa del objeto contiguo. El tamaño de la columna se ajustará automáticamente a los datos que contiene.

**Nota:** Si guarda la consulta después de cambiar el tamaño de las columnas o las filas, esta consulta siempre se visualizará con el nuevo formato.

#### Clasificación de resultados de consulta

Una vez ejecutada la consulta, puede clasificar los resultados por columna alfabéticamente.

En la vista de Resultados de una consulta, seleccione una columna y seleccione **Clasificación ascendente** en el menú **Resultados**.

Los resultados de consulta se clasifican por orden ascendente.

-o bien-

En la vista de Resultados de una consulta, seleccione una columna y seleccione **Clasificación descendente** en el menú **Resultados**.

Los resultados de consulta se clasifican por orden descendente.

**Nota:** Para aplicar una clasificación más compleja a la columna seleccionada, seleccione **Clasificar** en el menú **Resultados**.

## Reordenación de columnas

Puede cambiar el orden de las columnas en los resultados de consulta.

En la vista Resultados de una consulta, seleccione una columna y arrástrela hasta su nueva ubicación.

La columna aparecerá en el nuevo orden.

---

## Cómo aplicar formato a resultados de consulta

Es posible cambiar la fuente utilizada para visualizar consultas y resultados de consulta. La selección de fuentes dependerá de lo que se haya instalado en su sistema. Para obtener más información sobre cómo añadir fuentes, consulte el recurso de ayuda del sistema operativo.

**Nota:** Si guarda la consulta después de seleccionar una fuente nueva para visualizar los resultados de consulta, estos resultados siempre se visualizarán con la nueva fuente.

## Selección de la fuente de visualización de resultados de consulta

1. En la vista de Resultados, seleccione **Establecer fuente** en el menú **Resultados**. Se abrirá el recuadro de diálogo Fuente.
2. Seleccione la fuente y el tamaño de tipo para visualizar los resultados de la consulta y pulse en **Aceptar**. Los resultados de consulta se visualizan en el formato que ha especificado.

**Nota:** Pulse el botón en **Establecer como valor por omisión** para utilizar la fuente seleccionada como fuente por omisión para todos los resultados de consulta.

## Cómo aplicar formato a resultados de consulta numéricos

1. En la vista de Resultados, seleccione una columna que contenga valores numéricos y seleccione **Formato** en el menú **Resultados**. Se abrirá el recuadro de diálogo Formato.
2. Especifique el formato que desee aplicar y pulse en **Aceptar**.

**Nota:** Pulse en **Establecer como valor por omisión** para utilizar la fuente seleccionada como fuente por omisión para todos los resultados de consulta.

## Conversión del formato de los resultados de consulta en un formato

Puede convertir el formato de los resultados de consulta en un formato.

1. En el menú **Resultados**, seleccione **Visualizar informe**.

Se abrirá el recuadro de diálogo Seleccionar formato.

2. Seleccione Desde consulta y pulse en **Aceptar**.

El formato de los resultados de consulta se convertirá en un formato y se abrirá en una nueva ventana de formato.

---

## Cómo agrupar y agregar resultados de consulta

Puede agrupar y agregar los resultados de consulta y aplicarles formato de resumen.

### Agrupación de resultados de consulta

Puede agrupar los resultados de consulta con o sin información de resumen.

1. Seleccione la columna que desee agrupar.
2. En el menú **Resultados**, seleccione el tipo de agrupación que desee aplicar.

La columna se agrupará según la selección realizada.

### Cómo resumir resultados de consulta

Puede resumir los resultados de consulta por columna.

1. Seleccione la columna que desee agrupar.
2. En el menú **Resultados**, seleccione el tipo de resumen que desee aplicar.

La columna se resumirá según la selección realizada.

---

## Cómo guardar resultados de consulta y el formato

Puede guardar resultados de consulta y guardar el formato como un formato.

### Cómo guardar resultados de consulta como una tabla

Puede guardar los resultados de consulta como una tabla en un servidor de bases de datos.

1. En el menú **Resultados**, seleccione **Guardar en base de datos**.

Se abrirá el recuadro de diálogo Guardar datos.

2. Entre un propietario y un nombre de tabla y pulse en **Aceptar**.

Los resultados de consulta se guardarán como una tabla en la base de datos.

### Cómo guardar resultados de consulta en archivos

Puede guardar los resultados de consulta en un archivo en su PC o en un servidor de archivos.

1. En el menú **Resultados**, seleccione **Guardar en archivo**.

Se abrirá el recuadro de diálogo Exportar datos.

2. Especifique la ubicación en la cual desee guardar el archivo y las opciones de exportación, y pulse **Aceptar**.

Los resultados de consulta se guardarán en un archivo.

---

## Impresión de resultados de consulta

Puede realizar una presentación preliminar de los resultados de consulta e imprimirlos.

### Presentación preliminar de resultados de consulta

1. Abra y ejecute una consulta. Aparecerán los resultados de la consulta.
2. En el menú **Archivo**, seleccione **Preparar página**. Se abrirá el recuadro de diálogo Preparar página.
3. Efectúe los cambios que desee en el diseño de la página y pulse en **Aceptar**.
4. Pulse en el botón **Presentación preliminar** de la barra de herramientas:



Aparecerá una presentación preliminar de los resultados de consulta impresos.

### Impresión de resultados de consulta

1. Abra una consulta y active la vista de Resultados. Aparecerán los resultados de la consulta.
2. En el menú **Archivo**, seleccione **Preparar página**. Aparecerá el recuadro de diálogo Preparar página.
3. Efectúe los cambios que desee en el diseño de la página y pulse en **Aceptar**.
4. Pulse en el botón **Imprimir** de la barra de herramientas.



Se imprimirán los resultados de la consulta.

---

## Capítulo 9. Cómo trabajar con consultas estáticas

Una consulta estática es una consulta de SQL que se ha pasado anteriormente a un servidor de bases de datos y se ha vinculado en un paquete. Cuando se ejecuta una consulta estática, el servidor de bases de datos utiliza el texto de SQL vinculado en el paquete, en lugar del texto de SQL que aparece actualmente en la ventana de consulta. Las consultas estáticas son más eficientes en lo que se refiere a recursos que las consultas dinámicas, pero las consultas estáticas no se pueden editar.

---

### Consultas estáticas

Las consultas estáticas se crean a partir de consultas solicitud y consultas de SQL que ya existen de antemano.

#### Creación de consultas estáticas

1. En el menú **Consulta**, seleccione **Vincular paquete estático**. Se abrirá el recuadro de diálogo Vincular paquete estático.



2. Seleccione la pestaña **Paquete**, entre un nombre de paquete y un ID de colección y cambie cualquier otra opción según sus preferencias.
3. Si la consulta contiene variables de sustitución, seleccione la pestaña **Variables**. Sustituya las variables de sustitución por variables del sistema principal.
4. Pulse en **Aceptar**. Se vinculará la consulta estática.

**Nota:** Una vez que ha vinculado una consulta, también deberá guardarla en un archivo o en el servidor de bases de datos.

## Sustitución de variables de sustitución por variables del sistema principal

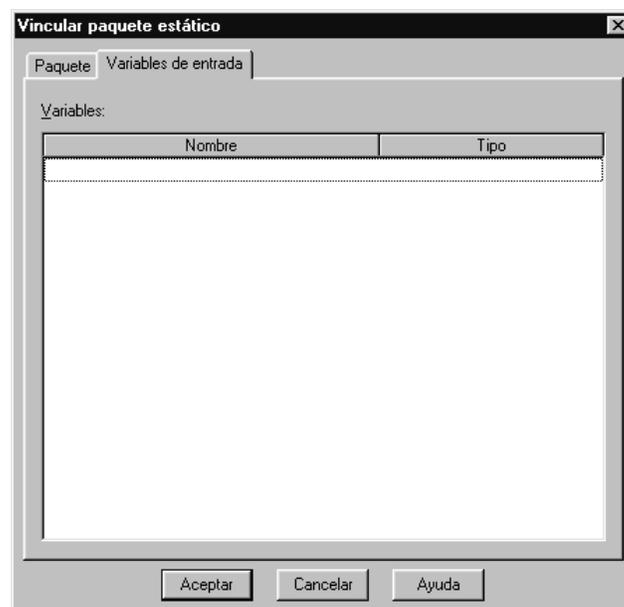
Al vincular un paquete, deberá especificar una variable del sistema principal para utilizarla en lugar de cada variable de sustitución en el texto de SQL. No obstante, una variable de sustitución no se puede sustituir siempre directamente por una variable del sistema principal. Las variables de sustitución proporcionan la sustitución de texto directa en el texto de consulta antes de que se envíe el texto al servidor de bases de datos. Las variables del sistema principal se envían como parte de la consulta al servidor de bases de datos. Consulte la documentación del servidor de bases de datos para conocer las reglas sobre cómo y dónde se pueden utilizar variables del sistema principal en las consultas.

Una vez que ha especificado la relación entre una variable de sustitución y una variable del sistema principal, QMF para Windows recuerda la relación y la sugiere en posteriores consultas, haciendo que la tarea de vinculación de paquetes resulte más simple.

Los tipos de datos válidos para las variables del sistema principal son:

- CHAR(n)
- VARCHAR(n)
- INTEGER
- SMALLINT
- FLOAT
- DECIMAL(p,s)
- DATE
- TIME
- TIMESTAMP

1. En el recuadro de diálogo Vincular paquete estático, seleccione la pestaña Variables de entrada.



2. Entre el tipo de variable para cada variable del sistema principal y pulse en **Aceptar**. Las variables de sustitución se convertirán en variables del sistema principal.

## **Ejecución de un consulta estática**

Las consultas estáticas se ejecutan igual que cualquier otra consulta. Consulte el apartado “Consultas de SQL” en la página “Consultas de SQL” en la página 7.



---

## Capítulo 10. Cómo trabajar con el Editor de tablas

Utilice el editor de tablas para buscar, añadir, editar o suprimir datos almacenados en las tablas sin tener que escribir sentencias de SQL.

---

### Editor de tablas

El editor de tablas le facilita mayor flexibilidad al editar y buscar datos.

#### Búsqueda de filas utilizando el editor de tablas

1. En el menú **Archivo**, seleccione **Editor de tablas**. Se abrirá el recuadro de diálogo Editor de tablas.



2. Especifique una tabla.

**Nota:** Puede utilizar patrones para seleccionar nombres de tabla en una lista de tablas coincidentes.

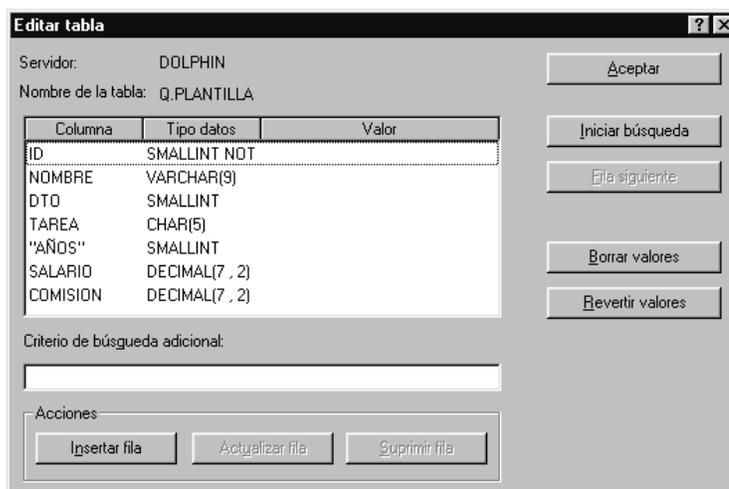
- Utilice el carácter de porcentaje (%) para comparar una serie de cualquier longitud que contenga cualquier carácter. Por ejemplo, para listar todas las tablas cuyo nombre empieza con la letra A, entre A%.
- Utilice el carácter de subrayado (\_) para comparar un carácter individual. Por ejemplo, para listar todas las tablas cuyo propietario tiene la letra A en la segunda posición, entre \_A%.

Una vez que haya entrado un patrón, pulse en **Listar tablas** y seleccione una tabla de la lista resultante.

3. Seleccione una Modalidad Guardar.

- Inmediato - La tabla se actualizará en el servidor de bases de datos justo después de cada cambio.
- Fin - La tabla se actualizará en el servidor de bases de datos después de que se haya terminado de entrar todos los cambios. Durante el tiempo que esté efectuando los cambios, otros usuarios no podrán realizar cambios en la

tabla.



Columna	Tipo datos	Valor
ID	SMALLINT NOT	
NOMBRE	VARCHAR(9)	
DTO	SMALLINT	
TAREA	CHAR(5)	
"AÑOS"	SMALLINT	
SALARIO	DECIMAL(7, 2)	
COMISION	DECIMAL(7, 2)	

4. Pulse el botón en **Editar**. Se abrirá el recuadro de diálogo Editar tabla.
5. Entre los valores que desea buscar en la columna Valor o escriba criterios de búsqueda en el campo Criterio de búsqueda adicional para especificar condiciones de búsqueda más complejas. Puede entrar cualquier predicado de SQL válido en el campo Criterio de búsqueda adicional.
6. Pulse en **Iniciar búsqueda**. En la columna Valor se visualizará la primera fila coincidente.

## Adición de una fila

1. En el recuadro de diálogo Editar tabla, entre la información para el nuevo registro.
2. Pulse en **Insertar fila**. La nueva fila se añadirá a la tabla.
3. Pulse en **Aceptar**. Se guardarán los cambios.

## Cambio de una fila

1. En el recuadro de diálogo Editar tabla, busque la fila que desea cambiar.
2. Pulse en **Fila siguiente** hasta que se visualice la fila que desea cambiar.
3. Edite los datos de la columna Valor y pulse en **Actualizar fila**. La fila se actualizará.
4. Pulse en **Aceptar**. Se guardarán los cambios.

## Supresión de una fila

1. En el recuadro de diálogo Editar tabla, busque la fila que desea suprimir.
2. Pulse en **Fila siguiente** hasta que se visualice la fila que desea suprimir.
3. Pulse en **Suprimir fila**. Se suprimirá la fila.
4. Pulse en **Aceptar**. Se guardarán los cambios.

---

## Edición de tablas desde la vista de resultados de consulta

Las tablas pueden editarse directamente desde la vista de resultados de consulta.

## Supresión de una fila desde la vista de resultados de consulta

En la vista de resultados de consulta es posible suprimir filas individuales de las tablas.

En la vista de resultados de consulta, seleccione una fila y seleccione **Suprimir** en el menú **Editar**. Se suprimirá la fila.

## Actualización de columnas desde la vista de resultados de consulta

En la vista de resultados de consulta también puede actualizar el contenido de columnas individuales.

En la vista de resultados de consulta, efectúe una doble pulsación en una casilla, entre un valor nuevo y pulse Intro. Se actualizará la tabla.

---

## DB2 Forms

Si ha instalado el componente de usuario DB2 Forms en la máquina, puede utilizarlo como un editor de tablas para tablas que no contengan datos LOB. Para obtener más información acerca de DB2 Forms, visite el Resource Center for DB2 Forms (Centro de recursos para DB2 Forms) en [www.rocketsoftware.com/db2forms](http://www.rocketsoftware.com/db2forms).



---

## Capítulo 11. Distribución de datos

Los datos pueden exportarse a otras bases de datos y aplicaciones.

---

### Exportación de datos

Puede exportar datos de QMF para Windows a otras aplicaciones del modo siguiente:

- Exporte los datos a un archivo de texto, CSV, IXF o HTML
- Guarde los resultados de consulta en una tabla
- Añada los resultados de consulta directamente en una hoja de cálculo de Microsoft Excel

### Exportación de datos a archivos

1. Mientras está viendo los resultados de la consulta, seleccione **Exportar datos** en el menú **Archivo**. Se abrirá el recuadro de diálogo Exportar datos.



2. Seleccione el Tipo de archivo de salida y pulse en el botón **Opciones**. En función del archivo de salida seleccionado, aparecerá el recuadro de diálogo Opciones de exportación de texto/DEL, Opciones de exportación de HTML, Opciones de exportación de IXF u Opciones de exportación de CSV.
  - Puede producir un archivo de texto con una extensión .TXT. Se trata de un archivo ASCII estándar con delimitadores de columna y serie opcionales (especificados en el recuadro de diálogo Opciones de exportación de texto/DEL).
  - Puede producir un archivo HTML con una extensión .HTM. Se trata de un archivo HTML que puede ver con cualquier examinador web. Todos los códigos HTML se generan automáticamente en el archivo; éste está listo para publicarse en el sitio web de Internet o intranet. Las opciones que elija en el recuadro de diálogo Opciones de exportación de HTML controlan el aspecto de los datos exportados.

- También puede producir un archivo .IXF. Una exportación en archivo IXF conserva toda la información de la base de datos, incluidos los encabezamientos de columna y los tipos de datos. Suele utilizarse para transferir información de una base de datos a otra.
  - Finalmente, otra posibilidad consiste en producir un archivo .CSV. Una exportación en archivo CSV es muy parecida a una exportación de texto; ya que se utiliza la coma como delimitador de columna. Este formato suele utilizarse en aplicaciones de hoja de cálculo.
3. Seleccione las opciones para el tipo de archivo de exportación seleccionado y pulse en **Aceptar**. El recuadro de diálogo Opciones se cerrará.
  4. Pulse en **Aceptar** en el recuadro de diálogo Exportar datos. Se exportarán los datos.

## Importación de datos

Los datos que se han guardado en un archivo IXF se pueden importar. Una vez importados los datos en una ventana de consulta, es posible guardarlos en un servidor de bases de datos, exportarlos a un archivo nuevo o utilizarlos en informes. Se soportan los archivos PCIXF y los archivos IXF de System/370 en modalidad de caracteres.

1. En el menú **Archivo**, seleccione **Importar datos**. Se abrirá el recuadro de diálogo Importar datos.



2. Seleccione el archivo que desea importar y pulse en **Aceptar**. Los datos importados se visualizarán en una ventana de consulta nueva.

## Cómo guardar datos en un servidor de bases de datos

Los resultados de consulta importados pueden guardarse en una tabla de bases de datos.

1. Mientras está los viendo resultados de la consulta, seleccione **Guardar datos** en el menú **Archivo**. Se abrirá el recuadro de diálogo Guardar datos.



2. Seleccione un servidor de bases de datos, entre un propietario y un nombre de tabla, elija las opciones que desee y pulse en **Aceptar**. Se guardarán los datos.

---

## Utilización del mandato Enviar a

QMF para Windows incluye un mandato Enviar a y un cliente de correo electrónico básico. Puede utilizar el mandato Enviar a junto con archivos de trabajos para planificar consultas y distribuir los resultados de las mismas.

1. En el menú **Archivo**, seleccione **Enviar a** y **Destinatario de correo de Internet**. Se abrirá el recuadro de diálogo Mensaje.
2. Especifique un destinatario del mensaje, un asunto, el texto del mensaje y pulse en **Siguiente**. Se abrirá el recuadro de diálogo Archivos adjuntos.
3. Añada o elimine los archivos adjuntos del mensaje y pulse en **Siguiente**. Se abrirá el recuadro de diálogo Enviar mensaje.
4. Especifique el nombre de su servidor de correo y pulse en **Finalizar**. El mensaje se envía.

---

## Utilización de la macro automática (Complementos) de Microsoft Excel

QMF para Windows incluye una macro automática para Microsoft Excel 7.0 o posterior. Estas macros automáticas permiten ejecutar QMF para Windows desde Excel y devolver los resultados de la consulta directamente en una hoja de cálculo. La macro automática apropiada se instala automáticamente si elige la opción de instalación "Típica" o si elige la opción de instalación "Personalizada" y selecciona la opción de macros automáticas Complementos de Microsoft Excel.

1. Pulse en el botón **QMF para Windows** de la barra de herramientas de Excel.



Se abrirá QMF para Windows.

2. En QMF para Windows, seleccione una consulta y ejecútela. Aparecerán los resultados de la consulta.
3. Seleccione los datos que desea devolver a Excel.

4. En el menú **Archivo**, seleccione **Devolver datos a Microsoft Excel**. Excel se abrirá y visualizará el recuadro de diálogo Macro automática de QMF para Windows.
5. Entre el rango de destino para los datos y pulse en **Aceptar**. Los datos se añadirán a la hoja de cálculo.

---

## Utilización de aplicaciones de ejemplo

Hay varias aplicaciones y soluciones integrales disponibles para QMF para Windows. Visite el sitio web de IBM en <http://www.ibm.com/qmf/> para obtener más información.

---

## Capítulo 12. Utilización del Centro de informes de QMF

El Centro de informes de QMF permite al usuario producir informes personalizados mediante consultas, formatos, procedimientos y tablas compartidas de QMF. Con un acceso rápido a estos objetos, podrá especificar sus preferencias de formato y producir informes personalizados que se podrán visualizar y manipular en una variedad de aplicaciones.

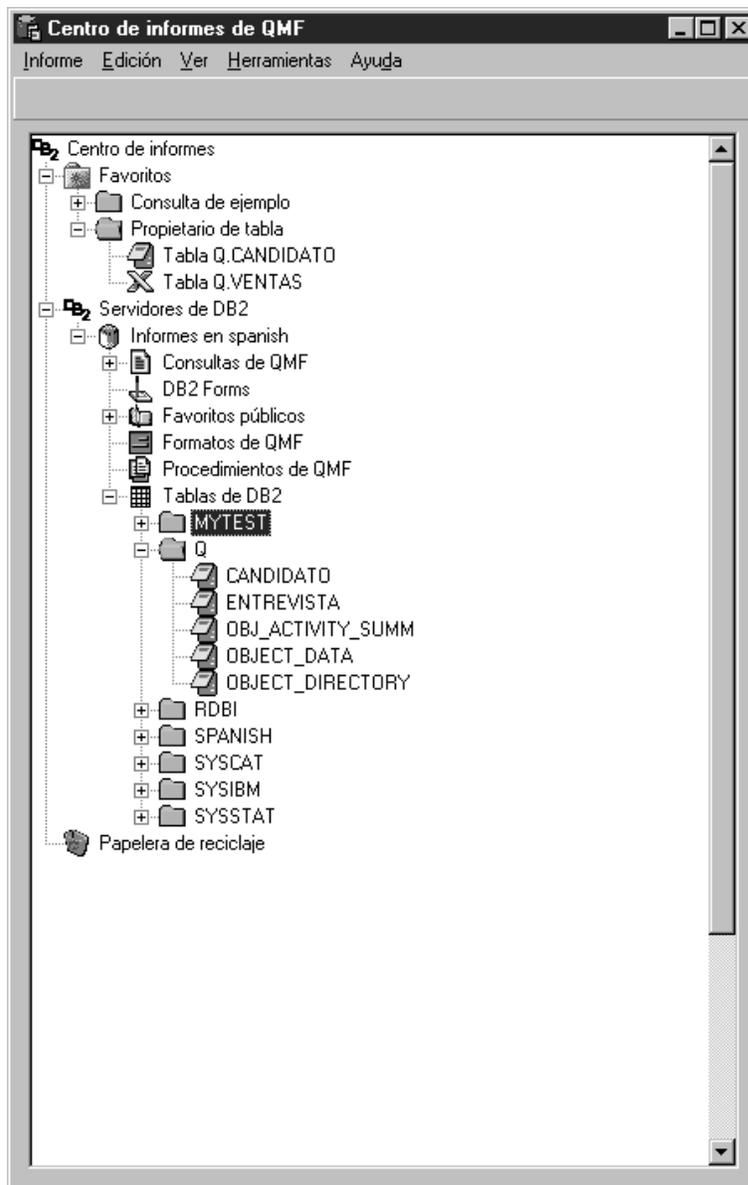
---

### Iniciación rápida del Centro de informes de QMF

- Efectúe una pulsación con el botón derecho del ratón en cualquier objeto o carpeta para activar las mismas opciones que están disponibles en los menús de la barra de herramientas.
- Pulse en el símbolo más (+) situado junto a cualquier carpeta para abrir el primer nivel de contenido. Mantenga pulsada la tecla MAYÚS mientras pulsa el símbolo más (+) para abrir todos los niveles bajo la carpeta.

### Ventana del Centro de informes de QMF

La ventana del Centro de informes de QMF contiene una estructura de árbol de los favoritos, servidores DB2, favoritos públicos y objetos disponibles, y una papelera de reciclaje.



Los objetos visualizados en esta ventana incluyen un icono que representa el tipo de aplicación con la que está asociada la salida del objeto.

## Conexión al servidor

1. Si no se visualizan nombres de servidor bajo Servidores DB2, pulse en el símbolo más (+).

2. Pulse en el símbolo más (+) situado junto a un servidor. Se abrirá el recuadro de diálogo Opciones de filtro.



3. Seleccione los tipos de objeto que desea visualizar, luego pulse en **Aceptar**. Se visualizarán los objetos disponibles en el servidor, agrupados por tipo de objeto.

---

## Cómo trabajar con Informes y Objetos

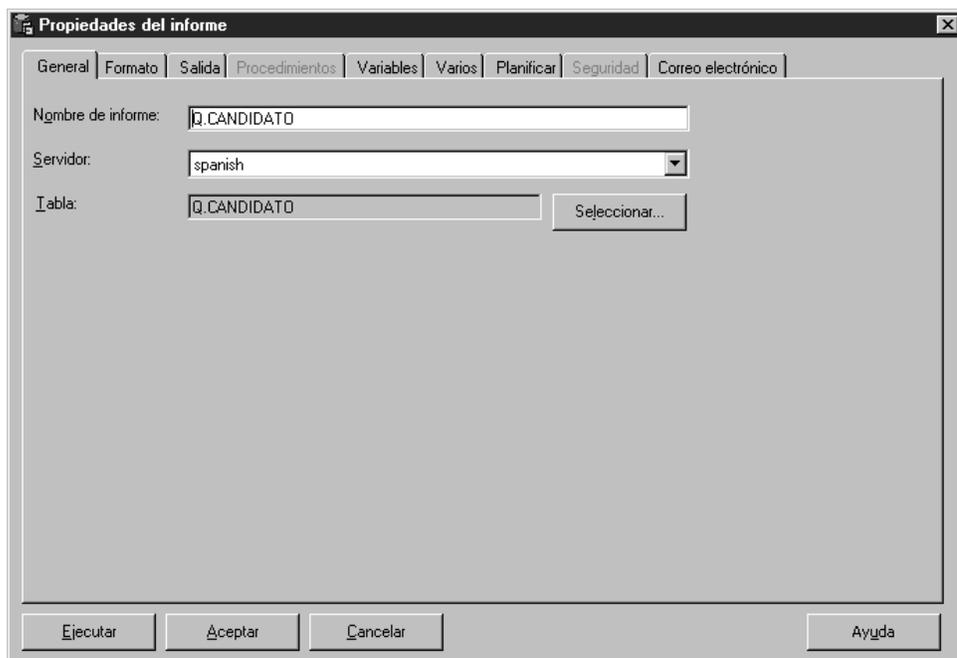
Los informes están basados en objetos de QMF. Todos los elementos contenidos en las carpetas Favoritos personales y públicos se consideran informes; puede manipular el formato y las opciones de visualización de estos elementos. Los elementos contenidos en las carpetas Favoritos enlazan con los objetos de QMF que residen en el servidor. No se modifica realmente un objeto de QMF, se modifica el enlace con el objeto al que se hace referencia como informe. Debido a que los informes se basan en objetos, las propiedades de los objetos también se aplican a los informes.

Puede crear informes a partir de objetos que residen en un servidor; sin embargo, no se guardan en el servidor. Este funcionamiento permite crear con rapidez informes puntuales. Sin embargo, después de crear un informe a partir de objetos que se encuentran en un servidor se le brindará la opción de guardar el informe en la carpeta Favoritos.

### Ejecución de informes

Puede ejecutar informes de la carpeta Favoritos o de objetos ubicados en el servidor.

1. Con el informe u objeto seleccionado, elija *Propiedades* del menú Informe. Se abrirá el recuadro de diálogo Propiedades del informe.



2. Defina las propiedades, si lo desea.
3. Pulse en el botón **Ejecutar**. El informe se procesa y, si ha seleccionado la opción *Ver informe después de su publicación* en el recuadro de diálogo Propiedades del informe: Salida, el informe se visualizará en la aplicación que ha especificado.

También puede ejecutar de manera rápida un informe mediante cualquiera de los métodos siguientes:

- Seleccione el informe, luego elija *Ejecutar* del menú Informe.
- Efectúe una pulsación con el botón derecho del ratón en el informe, luego elija *Ejecutar*.
- Efectúe una doble pulsación en el nombre del informe.

---

## Cómo trabajar con carpetas y favoritos

Las carpetas se utilizan para agrupar informes y objetos de QMF; las carpetas se nombran según los nombres del propietario de los objetos. Puede realizar las mismas operaciones en las carpetas que las que realiza en los informes, por ejemplo, ejecutar informes y definir propiedades de informe. La realización de estas operaciones en una carpeta aplica la operación a todos los informes contenidos en la carpeta. Por ejemplo, si desea ejecutar de manera consecutiva cada informe contenido en una carpeta, seleccione la carpeta, luego seleccione **Ejecutar** del menú Informe.

El Centro de informes de QMF contiene dos carpetas de nivel superior en las que puede almacenar informes. Las carpetas contienen informes que enlazan con objetos que se encuentran en servidores; los propios objetos no están contenidos en la carpeta Favoritos. La carpeta Favoritos personales reside localmente (en el PC), de forma que usted es el único usuario que puede acceder a esta carpeta y al contenido de la misma. La carpeta Favoritos públicos reside en el servidor y pueden acceder a ella todos los usuarios autorizados. Se puede tener acceso a

varias carpetas Favoritos públicos, en función de los límites de recursos, aunque nunca habrá más de una carpeta Favoritos públicos en cada servidor.

Cuando copia objetos de QMF a las carpetas Favoritos, las carpetas se renombran automáticamente para incluir el tipo de objeto y el nombre del propietario. Cuando copia un tipo de objeto entero (es decir, carpeta de objetos del mismo tipo) de un servidor (por ejemplo, todas las consultas), el nombre del servidor también se incluye en el nuevo nombre de carpeta.

## Cómo añadir informes a favoritos

Puede añadir objetos o informes de un servidor a su carpeta Favoritos personales o en la carpeta Favoritos públicos en un servidor (siempre que el administrador del sistema le haya otorgado permiso).

### Para añadir informes a Favoritos personales:

Con el informe u objeto seleccionado, elija *Añadir a favoritos* del menú Informe o arrastre el informe u objeto a su carpeta Favoritos personales. Se añadirá un informe en la parte superior de su carpeta Favoritos personales con la convención de nomenclatura siguiente: TipoobjetoNOMBREPROPIETARIO.NOMBREOBJETO.

### Para añadir informes a Favoritos públicos:

Arrastre el objeto de QMF o informe a la carpeta Favoritos públicos en un servidor. Puede añadir informes de la carpeta Favoritos personales o de cualquier otro servidor.

**Nota:** Al añadir a Favoritos públicos o al modificar informes en Favoritos públicos, debe seleccionar *Guardar cambios en Favoritos públicos* del menú Informe antes de que las actualizaciones se guarden en el servidor.

Para obtener más información acerca de cómo utilizar el Centro de informes de QMF, consulte el sistema de ayuda en línea.



---

## Capítulo 13. Utilización de la API de QMF para Windows

Puede crear aplicaciones personalizadas mediante la interfaz de programación de aplicaciones (API) de QMF para Windows.

---

### Cómo controlar QMF para Windows mediante la API

Los pasos siguientes proporcionan una visión general de cómo el usuario trabaja con la API para controlar QMF para Windows.

1. Cree una instancia de objeto de la API de QMF para Windows. Si utiliza Microsoft Visual Basic, añada una referencia a la biblioteca de tipo de QMF para Windows, qmfwin.tlb. Luego utilice la sentencia Dim:

```
Dim QMFWin As New QMFWin
```

O la sentencia CreateObject:

```
Dim QMFWin As Object
```

```
Set QMFWin = CreateObject ("QMFWin.Interface")
```

- Nota:** Si utiliza un entorno de desarrollo diferente, consulte la documentación del producto para obtener información sobre cómo completar este paso.
2. Seleccione el servidor DB2 que desee utilizar y llame a InitializeServer() para inicializar una conexión a la base de datos.

**Nota:** No podrá inicializar un servidor DB2 hasta que DB2 valide un ID de usuario y una contraseña. Puede especificar que QMF para Windows solicite el ID de usuario y la contraseña, o bien que la aplicación solicite esta información y transferirla después en forma de parámetros en la llamada a la función InitializeServer().

3. Seleccione la consulta que desee ejecutar utilizando InitializeQuery(). Si la consulta contiene variables, utilice la función SetVariable() para establecer los valores de las variables.
4. Abra o ejecute la consulta. Utilice la función Open() para abrir el cursor de la consulta para las sentencias SELECT y utilice la función Execute() para ejecutar el SQL para las sentencias no SELECT.
5. Si la consulta es una sentencia SELECT, recupere filas de datos llamando repetidamente a FetchNextRow(). Para recuperar más de una fila al mismo tiempo, utilice FetchNextRows(), o utilice CompleteQuery() para indicar a QMF para Windows que recupere todas las filas.
6. Si la consulta es un sentencia SELECT, cierre la consulta mediante la función Close().
7. Termine la unidad de trabajo utilizando las funciones Commit() o Rollback().

### Bloqueo de llamadas

Todas las funciones de la API de QMF para Windows son síncronas. Esto significa que se bloquean o no vuelven hasta que se ha completado la acción solicitada de la base de datos. Esta implementación es adecuada porque simplifica la programación en la aplicación cliente. Sin embargo, si la aplicación cliente es de una sola hebra, no podrá responder a la entrada de usuario ni actualizar la pantalla mientras esté esperando a que vuelva una función de la API de QMF para Windows.

La API de QMF para Windows responde a las llamadas de función de un cliente de una en una. Si la aplicación cliente tiene múltiples hebras, deberá:

- esperar a que se complete una llamada de función antes de efectuar otra o
- crear múltiples instancias de la API de QMF para Windows (una para cada hebra que utiliza la API).

## Conexión a la base de datos

Cada instancia de objeto de la API de QMF para Windows crea y utiliza una conexión individual a la base de datos para toda la actividad de la base de datos que está sujeta a una retrotracción o confirmación subsiguiente, incluidas la apertura de una consulta, la recuperación de datos y la ejecución de sentencias de SQL.

Si crea más de una consulta en una instancia determinada del objeto de la API de QMF para Windows llamando a `InitializeQuery()` dos o más veces, todas las consultas comparten la misma conexión individual.

La API de QMF para Windows responde a las llamadas de función de un cliente de una en una. Si la aplicación cliente tiene múltiples hebras, deberá:

- `DeleteQMFObject()`
- `GetQMFObjectInfo()`
- `GetQMFObjectInfoEx()`
- `GetQMFObjectList()`
- `GetQMFObjectListEx()`
- `GetQMFQueryText()`
- `SaveQMFQuery()`

QMF para Windows crea y utiliza una segunda conexión a la base de datos con el fin de manejar la actividad administrativa de la base de datos (por ejemplo, recuperar información de QMF). Esta segunda conexión es necesaria para soportar un mecanismo de retrotracción y confirmación coherente para las aplicaciones cliente.

El objeto de la API de QMF para Windows maneja automáticamente estas conexiones a la base de datos. Sin embargo, si el administrador del sistema ha establecido un límite para el número de conexiones permitidas, recuerde que cada instancia del objeto de la API de QMF para Windows puede utilizar dos conexiones.

---

## Consulta de la API

Esta consulta lista todos los mandatos disponibles para crear aplicaciones mediante la API de QMF para Windows.

### **AddDecimalHostVariable()**

short AddDecimalHostVariable(long *IDConsulta*, short *Tipo*, short *Precisión*, short *Escala*, const VARIANT& *Valor*)

#### **Descripción**

Esta función aplica los datos de *Valor* a una variable de la sentencia de SQL estático inicializada con *IDConsulta*. Esta función se llama para cada variable de la

sentencia. QMF para Windows no intenta emparejar valores con variables, de modo que esta función deberá llamarse en el mismo orden que las variables de la sentencia SQL.

### Parámetros

Nombre	Descripción
<i>IDConsulta</i>	El ID de la consulta devuelto desde <code>InitializeStaticQuery()</code> .
<i>Tipo</i>	Tipo de datos SQL del valor que se debe pasar al servidor de bases de datos. Este valor influye en la conversión de <i>Valor</i> de un tipo de datos VARIANT al valor pasado actualmente. El único valor válido para <code>AddDecimalHostVariable()</code> es 484 (RSDT_DECIMAL).
<i>Precisión</i>	La precisión del valor decimal.
<i>Escala</i>	La escala del valor decimal.
<i>Valor</i>	El valor de datos que se va a sustituir en la sentencia. Para especificar un valor nulo, el tipo de variante debería estar establecido en VT_EMPTY.

### Valor devuelto

Cero indica que la función se ha completado correctamente; un valor distinto de cero significa que ha habido un error. Si el valor devuelto es distinto de cero, puede llamar a `GetLastErrorString()` o `GetLastErrorType()` para obtener información adicional del error.

## AddHostVariable()

short AddHostVariable(long *IDConsulta*, short *Tipo*, const VARIANT& *Valor*)

### Descripción

Esta función aplica los datos de *Valor* a una variable de la sentencia de SQL estático inicializada con *IDConsulta*. Deberá llamar a esta función para cada variable de la sentencia. QMF para Windows no intenta comparar valores con variables, de modo que esta función deberá llamarse en el mismo orden que las variables de la sentencia SQL.

### Parámetros

Nombre	Descripción
<i>IDConsulta</i>	ID de la consulta devuelto desde <code>InitializeStaticQuery()</code> .
<i>Tipo</i>	Tipo de datos SQL del valor que se debe pasar al servidor de bases de datos. Este valor influye en la conversión de <i>Valor</i> de un tipo de datos VARIANT al valor pasado actualmente.
<i>Valor</i>	Valor de datos que se va a sustituir en la sentencia. Para especificar un valor nulo, el tipo de variante debería estar establecido en VT_EMPTY.

Los valores válidos para *Tipo* incluyen:

Valor	Significado
384 (RSDT_DATE)	Fecha
388 (RSDT_TIME)	Hora
392 (RSDT_TIMESTAMP)	Indicación de la hora
448 (RSDT_VARCHAR)	Serie de caracteres de longitud variable

452 (RSDT_CHAR)	Serie de caracteres
464 (RSDT_VARGRAPHIC)	Gráfico de longitud variable
468 (RSDT_GRAPHIC)	Gráfico
480 (RSDT_FLOAT)	Número de coma flotante
496 (RSDT_INTEGER)	Entero de 4 bytes
500 (RSDT_SMALLINT)	Entero de 2 bytes

### Valor devuelto

Cero indica que la función se ha completado correctamente; un valor distinto de cero significa que ha habido un error. Si el valor devuelto es distinto de cero, puede llamar a `GetLastErrorString()` o `GetLastErrorType()` para obtener información adicional del error.

## BindDecimalHostVariable()

short `BindDecimalHostVariable(BSTR NombreColección, BSTR NombrePaquete, short NúmeroSección, short Número, BSTR Nombre, short TipoDatos, short Precisión, short Escala)`

### Descripción

Esta función vincula una variable de la sección especificada. Incluya el texto ":H" en el texto de SQL como un espacio reservado para una variable del sistema principal. Para cada variable decimal de sistema principal del texto de SQL, deberá llamar a `BindDecimalHostVariable()` para especificar información sobre la variable.

### Parámetros

Nombre	Descripción
<i>NombreColección</i>	ID de colección del paquete que se desea vincular.
<i>NombrePaquete</i>	Nombre del paquete que se desea vincular.
<i>NúmeroSección</i>	Número de sección de la sentencia dentro de la colección y el paquete que desea vincular.
<i>Número</i>	Identificador de la variable que se desea vincular. La primera variable de la sentencia SQL es 0, etc.
<i>Nombre</i>	Lo utiliza el servidor de bases de datos para diagnósticos. QMF para Windows no valida ni requiere este valor.
<i>TipoDatos</i>	Tipo de datos SQL de la variable. El único valor válido para <code>BindDecimalHostVariable()</code> es 484 (RSDT_DECIMAL).
<i>Precisión</i>	Precisión del valor decimal.
<i>Escala</i>	Escala del valor decimal.

### Valor devuelto

Cero indica que la función se ha completado correctamente; un valor distinto de cero significa que ha habido un error. Si el valor devuelto es distinto de cero, puede llamar a `GetLastErrorString()` o `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` o `GetLastSQLState()` para obtener información adicional del error.

## BindHostVariable()

short `BindHostVariable(BSTR NombreColección, BSTR NombrePaquete, short NúmeroSección, short Número, BSTR Nombre, short TipoDatos, short Longitud)`

## Descripción

Esta función vincula una variable de la sección especificada. Incluya el texto ":H" en el texto de SQL como un espacio reservado para una variable del sistema principal. Para cada variable de sistema principal del texto de SQL, deberá llamar a BindHostVariable() para especificar información sobre la variable.

## Parámetros

Nombre	Descripción
<i>NombreColección</i>	ID de colección del paquete que se desea vincular.
<i>NombrePaquete</i>	Nombre del paquete que se desea vincular.
<i>NúmeroSección</i>	Número de sección de la sentencia dentro de la colección y el paquete que desea vincular.
<i>Número</i>	Identificador de la variable que se desea vincular. La primera variable de la sentencia SQL es 0, etc.
<i>Nombre</i>	Lo utiliza el servidor de bases de datos para diagnósticos. QMF para Windows no valida ni requiere este valor.
<i>TipoDatos</i>	Tipo de datos SQL de la variable.
<i>Longitud</i>	Longitud de la variable.

Los valores válidos para *TipoDatos* incluyen:

Valor	Significado
384 (RSDT_DATE)	Fecha
388 (RSDT_TIME)	Hora
392 (RSDT_TIMESTAMP)	Indicación de la hora
448 (RSDT_VARCHAR)	Serie de caracteres de longitud variable
452 (RSDT_CHAR)	Serie de caracteres
464 (RSDT_VARGRAPHIC)	Gráfico de longitud variable
468 (RSDT_GRAPHIC)	Gráfico
480 (RSDT_FLOAT)	Número de coma flotante
484 (RSDT_DECIMAL)	Decimal
496 (RSDT_INTEGER)	Entero de 4 bytes
500 (RSDT_SMALLINT)	Entero de 2 bytes

## Valor devuelto

Cero indica que la función se ha completado correctamente; un valor distinto de cero significa que ha habido un error. Si el valor devuelto es distinto de cero, puede llamar a GetLastErrorString() o GetLastErrorType(), GetLastErrorCode(), GetLastErrorError() o GetLastErrorState() para obtener información adicional del error.

## BindSection()

short BindSection(BSTR *NombreColección*, BSTR *NombrePaquete*, short *NúmeroSección*, BSTR *TextoSQL*)

## Descripción

Esta función establece el texto de SQL que se debe utilizar en el número de sección especificado de la colección y del paquete durante la vinculación.

## Parámetros

Nombre	Descripción
NombreColección	ID de colección del paquete que se desea vincular.
NombrePaquete	Nombre del paquete que se desea vincular.
NúmeroSección	Número de sección de la sentencia dentro de la colección y el paquete que desea vincular.
TextoSQL	Texto de SQL para la sentencia que se desea vincular.

### Valor devuelto

Cero indica que la función se ha completado correctamente; un valor distinto de cero significa que ha habido un error. Si el valor devuelto es distinto de cero, puede llamar a `GetLastErrorString()` o `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` o `GetLastSQLState()` para obtener información adicional del error.

## CancelBind()

short CancelBind(BSTR *NombreColección*, BSTR *NombrePaquete*)

### Descripción

Esta función cancela una función de vinculación inicializada anteriormente. Se libera toda la información relacionada con el paquete nombrado.

### Parámetros

Nombre	Descripción
NombreColección	ID de colección del paquete que se desea vincular.
NombrePaquete	Nombre del paquete que se desea vincular.

### Valor devuelto

Cero indica que la función se ha completado correctamente; un valor distinto de cero significa que ha habido un error. Si el valor devuelto es distinto de cero, puede llamar a `GetLastErrorString()` o `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` o `GetLastSQLState()` para obtener información adicional del error.

## ChangePassword()

short ChangePassword(BSTR *ContraseñaNueva*)

### Descripción

Esta función cambia la contraseña para el ID de usuario especificado previamente en la llamada a `InitializeServer()`.

**Nota:** No todos los tipos de servidores de bases de datos dan soporte a cambios de contraseña. Si el servidor especificado en la llamada a `InitializeServer()` no da soporte al cambio de contraseña, se devuelve un error y la contraseña no se cambia.

### Parámetros

Nombre	Descripción
ContraseñaNueva	La nueva contraseña.

### Valor devuelto

Cero indica que la función se ha completado correctamente; un valor distinto de cero significa que ha habido un error. Si el valor devuelto es distinto de cero, puede llamar a `GetLastErrorString()` o `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` o `GetLastSQLState()` para obtener información adicional del error.

## ClearList()

short ClearList(short *Tipo*)

### Descripción

Esta función reinicializa la lista interna especificada por el parámetro *Tipo*.

### Parámetros

Nombre	Descripción
<i>Tipo</i>	El valor RSL_SERVER o RSL_QUERY.

### Valor devuelto

Cero indica que la función se ha completado correctamente; RS\_ERROR\_OUTOFRANGE no es satisfactorio.

### Temas relacionados

Open()

## Close()

short Close(long *IDConsulta*)

### Descripción

Esta función cierra una consulta e invalida el *IDConsulta*. Si existe un cursor abierto para la consulta, éste se cierra liberando la base de datos para otros usuarios. Esta función no termina la conexión con el servidor de bases de datos. Dado que la conexión permanece abierta, no se lleva a cabo confirmación o retrotracción alguna.

**Nota:** El nombre de esta función está en conflicto con la palabra clave Close de Microsoft Access 2.0. Si está utilizando MS Access 2.0, ponga el nombre de función entre corchetes [ ].

### Parámetros

Nombre	Descripción
<i>IDConsulta</i>	ID de la consulta devuelto desde <code>InitializeQuery()</code> .

### Valor devuelto

Cero indica que la función se ha completado correctamente; un valor distinto de cero significa que ha habido un error. Si el valor devuelto es distinto de cero, puede llamar a `GetLastErrorString()` o `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` o `GetLastSQLState()` para obtener información adicional del error.

### Temas relacionados

Execute()

Open()

## Commit()

short Commit(long *IDConsulta*)

### Descripción

Esta función confirma los cambios efectuados en la unidad de trabajo actual, finaliza la unidad de trabajo actual, cierra los cursores e invalida todos los ID de consulta pendientes.

**Nota:** El nombre de esta función está en conflicto con la palabra clave Commit de Microsoft Access 2.0. Si está utilizando MS Access 2.0, ponga el nombre de función entre corchetes [ ].

### Valor devuelto

Cero indica que la función se ha completado correctamente; un valor distinto de cero significa que ha habido un error. Si el valor devuelto es distinto de cero, puede llamar a GetLastErrorString() o GetLastErrorType(), GetLastErrorSQLCode(), GetLastErrorSQLError() o GetLastErrorSQLState() para obtener información adicional del error.

### Temas relacionados

Rollback()

## CompleteQuery()

short CompleteQuery(long *IDConsulta*)

### Descripción

Esta función recupera todas las filas de un conjunto de resultados y las almacena internamente en QMF para Windows. Si existe un cursor abierto para la consulta, éste se cierra liberando la base de datos para otros usuarios. Utilice FetchNextRow() o FetchNextRows() para recuperar las filas. Llame a Close() cuando haya terminado con esta consulta.

### Parámetros

Nombre	Descripción
<i>IDConsulta</i>	ID de la consulta devuelto desde InitializeQuery().

### Valor devuelto

Cero indica que la función se ha completado correctamente; un valor distinto de cero significa que ha habido un error. Si el valor devuelto es distinto de cero, puede llamar a GetLastErrorString() o GetLastErrorType(), GetLastErrorSQLCode(), GetLastErrorSQLError() o GetLastErrorSQLState() para obtener información adicional del error.

## CopyToClipboard()

short CopyToClipboard(long *IDConsulta*, long *PrimeraFila*, long *PrimeraCol*, long *ÚltimaFila*, long *ÚltimaCol*, BOOL *IncluirCabecerasCol*, [VARIANT *FormatoFechaHora*])

### Descripción

Esta función copia el rango especificado de filas y columnas en el portapapeles. Si no ha recuperado datos de fila para todas las filas que desea copiar en el portapapeles, llame a CompleteQuery() antes de llamar a esta función. Se devolverá un mensaje de error si intenta copiar filas que no se han recuperado de la base de datos.

## Parámetros

Nombre	Descripción
<i>IDConsulta</i>	ID de la consulta devuelto desde InitializeQuery().
<i>PrimeraFila</i>	Primera fila que desea incluir en la copia.
<i>PrimeraCol</i>	Primera columna que desea incluir en la copia.
<i>ÚltimaFila</i>	Última fila que desea incluir en la copia o -1 si se incluyen todas las filas.
<i>ÚltimaCol</i>	Última columna que desea incluir en la copia o -1 si se incluyen todas las columnas.
<i>IncluirCabecerasCol</i>	Utilice un valor distinto de cero para incluir las cabeceras de columna en la primera fila y cero para no incluirlas.
<i>FormatoFechaHora</i>	Opcionalmente, formato que se va a utilizar para los valores de fecha y hora. Los valores válidos son 0 (formato ISO), 1 (formato USA), 2 (formato EUR), 3 (formato JIS) o 4 (formato de panel de control de Windows). El valor por omisión es 4.

**Nota:** El valor de la primera fila en un conjunto de resultados es 0 y el valor de la última fila es el número total de filas menos uno. El valor de la primera columna en un conjunto de resultados es 0 y el valor de la última columna es el número total de columnas menos uno.

### Valor devuelto

Cero indica que la función se ha completado correctamente; un valor distinto de cero significa que ha habido un error. Si el valor devuelto es distinto de cero, puede llamar a `GetLastErrorString()` o `GetLastErrorType()` para obtener información adicional del error. Si el conjunto de resultados está vacío o no se ha recuperado ninguna fila de la base de datos, se devuelve un valor distinto de cero a no ser que *PrimeraFila*=0 y *ÚltimaFila*=1. En este caso, se devuelve cero y se copia una serie vacía en el portapapeles.

## DeleteQMFObject()

short DeleteQMFObject(BSTR *PropietarioYNombre*)

### Descripción

Esta función suprime un objeto de QMF (consulta, formato, procedimiento o tabla).

### Parámetros

Nombre	Descripción
<i>PropietarioYNombre</i>	Serie que contiene el propietario y el nombre, separados por un punto, del objeto que se desea suprimir. Por ejemplo, Juan.Consulta2

### Valor devuelto

Cero indica que la función se ha completado correctamente; un valor distinto de cero significa que ha habido un error. Si el valor devuelto es distinto de cero, puede llamar a `GetLastErrorString()` o `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` o `GetLastSQLState()` para obtener información adicional del error.

## EndBind()

short EndBind(BSTR *NombreColección*, BSTR *NombrePaquete*)

### Descripción

Esta función completa el proceso de vinculación de un paquete SQL estático. La llamada a esta función hace que QMF para Windows envíe la información completa del paquete actual a la base de datos para su proceso.

### Parámetros

Nombre	Descripción
<i>NombreColección</i>	Nombre de colección utilizado en la llamada a StartBind() anterior.
<i>NombrePaquete</i>	Nombre de paquete utilizado en la llamada a StartBind() anterior.

### Valor devuelto

Cero indica que la función se ha completado correctamente; un valor distinto de cero significa que ha habido un error. Si el valor devuelto es distinto de cero, puede llamar a GetLastErrorString() o GetLastErrorType(), GetLastErrorSQLCode(), GetLastErrorSQLError() o GetLastErrorSQLState() para obtener información adicional del error.

## Execute()

short Execute(long *IDConsulta*)

### Descripción

Esta función ejecuta una sentencia SQL que utiliza un verbo de SQL distinto de SELECT. Utilice Execute() cuando la sentencia no devuelva ningún resultado. Para sentencias que devuelven resultados, ejecute ExecuteEx(). Para sentencias que utilizan el verbo SELECT, utilice Open() en lugar de Execute() o ExecuteEx(). Para determinar el verbo utilizado por una consulta, llame a GetQueryVerb().

**Nota:** El nombre de esta función está en conflicto con la palabra clave Execute de Microsoft Access 2.0. Si está utilizando MS Access 2.0, ponga el nombre de función entre corchetes [ ].

### Parámetros

Nombre	Descripción
<i>IDConsulta</i>	ID de la consulta devuelto desde InitializeQuery().

### Valor devuelto

Cero indica que la función se ha completado correctamente; un valor distinto de cero significa que ha habido un error. Si el valor devuelto es distinto de cero, puede llamar a GetLastErrorString() o GetLastErrorType(), GetLastErrorSQLCode(), GetLastErrorSQLError() o GetLastErrorSQLState() para obtener información adicional del error.

### Temas relacionados

Execute()

Open()

## ExecuteEx()

short ExecuteEx(long *IDConsulta*, VARIANT\* *Resultado*)

## Descripción

Esta función ejecuta una sentencia SQL que utiliza un verbo de SQL distinto de SELECT. Utilice ExecuteEx() cuando la sentencia devuelva resultados, por ejemplo, con una sentencia SELECT INTO. Para sentencias que no devuelven ningún resultado, utilice Execute(). Para sentencias que utilizan el verbo SELECT, utilice Open() en lugar de Execute() o ExecuteEx(). Para determinar el verbo utilizado por una consulta, llame a GetQueryVerb().

## Parámetros

Nombre	Descripción
<i>IDConsulta</i>	ID de la consulta devuelto desde InitializeQuery().
<i>Resultado</i>	<p>Puntero de una variable VARIANT donde se almacenará el resultado. El resultado es una matriz (tipo de variante VT_ARRAY   VT_VARIANT) que contiene un valor para cada columna del resultado.</p> <p>Cada valor se especifica en su tipo de datos nativo o en el tipo de datos de variante más cercano. Los tipos de valores devueltos con soporte son: serie (tipo de variante VT_BSTR), flotante (tipo de variante VT_R4), doble (tipo de variante VT_R8), corto (tipo de variante VT_I2), largo (tipo de variante VT_I4) y binario (tipo de variante VT_UI1   VT_ARRAY).</p> <p>Deberá inicializar la variable VARIANT correctamente antes de llamar a esta función. Visual Basic efectúa esta acción automáticamente. Los programadores de Visual C++ deben llamar a VariantInit().</p>

## Valor devuelto

Cero indica que la función se ha completado correctamente; un valor distinto de cero significa que ha habido un error. Si el valor devuelto es distinto de cero, puede llamar a GetLastErrorString() o GetLastErrorType(), GetLastErrorCode(), GetLastErrorError() o GetLastErrorSQLState() para obtener información adicional del error.

## ExecuteStored Procedure()

```
short ExecuteStoredProcedure(long IDConsulta,[VARIANT vaConfirmarOK],  
[VARIANT vaMáxConjuntosResultados], [VARIANT vaNombresColumnas], [VARIANT  
vaEtiquetasColumnas], [VARIANT vaComentariosColumnas])
```

## Descripción

Esta función ejecuta una sentencia SQL que utiliza un verbo CALL de SQL para ejecutar un procedimiento almacenado en el servidor de bases de datos. Utilice ExecuteStoredProcedure() cuando el procedimiento almacenado no devuelva resultados (en lugar de o además de los conjuntos de resultados). Para procedimientos almacenados que devuelvan resultados, utilice ExecuteStoredProcedureEx().

Para inicializar un procedimiento almacenado para su ejecución con ExecuteStoredProcedure(), primero llame a InitializeQuery() especificando una sentencia SQL que utilice la sentencia CALL. El nombre del procedimiento almacenado se debe especificar como literal en la sentencia CALL. Los parámetros especificados en la sentencia CALL (constantes o no) se ignoran. En su lugar, utilice AddHostVariable() para especificar las variables de entrada y salida.

## Parámetros

Nombre	Descripción
<i>IDConsulta</i>	ID de la consulta devuelto desde InitializeQuery(). El texto de SQL para la consulta debe especificar una sentencia CALL.
<i>vaConfirmarOK</i>	Valor booleano opcional que especifica si el procedimiento almacenado puede confirmar la unidad de trabajo o si esta operación se debe restringir. El valor por omisión es True.
<i>vaMáxConjuntosResultados</i>	Valor numérico opcional que especifica el número máximo de conjuntos de resultados que se debe permitir que el procedimiento almacenado devuelva. Especifique cero si no desea que el procedimiento almacenado devuelva ningún conjunto de resultados o si el servidor de bases de datos no da soporte a la devolución de conjuntos de resultados por parte de procedimientos almacenados a través de DRDA.
<i>vaNombresColumnas</i>	Valor booleano opcional que especifica si la base de datos debe devolver los nombres de las columnas para las columnas de cada conjunto de resultados devuelto.
<i>vaEtiquetasColumnas</i>	Valor booleano opcional que especifica si la base de datos debe devolver las etiquetas de las columnas de cada conjunto de resultados devuelto.
<i>vaComentariosColumnas</i>	Valor booleano opcional que especifica si la base de datos debe devolver comentarios para las columnas de cada conjunto de resultados devuelto.

### Valor devuelto

Cero indica que la función se ha completado correctamente; un valor distinto de cero significa que ha habido un error. Si el valor devuelto es distinto de cero, puede llamar a GetLastErrorString() o GetLastErrorType(), GetLastErrorSQLCode(), GetLastErrorSQLError() o GetLastErrorSQLState() para obtener información adicional del error.

## ExecuteStored ProcedureEx()

```
short ExecuteStoredProcedureEx(long IDConsulta,VARIANT* Resultado,[VARIANT vaConfirmarOK], [VARIANT vaMáxConjuntosResultados], [VARIANT vaNombresColumnas], [VARIANT vaEtiquetasColumnas], [VARIANT vaComentariosColumnas])
```

### Descripción

Esta función ejecuta una sentencia SQL que utiliza un verbo CALL de SQL para ejecutar un procedimiento almacenado en el servidor de bases de datos. Utilice ExecuteStoredProcedureEx() cuando el procedimiento almacenado devuelva resultados (en lugar de o además de los conjuntos de resultados). Para procedimientos almacenados que no devuelvan resultados, utilice ExecuteStoredProcedure().

Para inicializar un procedimiento almacenado para su ejecución con ExecuteStoredProcedureEx(), primero llame a InitializeQuery() especificando una sentencia SQL que utilice la sentencia CALL. El nombre del procedimiento almacenado se debe especificar como literal en la sentencia CALL. Los parámetros especificados en la sentencia CALL (constantes o no) se ignoran. En su lugar, utilice AddHostVariable() para especificar las variables de entrada y salida.

Si el procedimiento almacenado devuelve conjuntos de resultados, llame a `GetStoredProcedureResultSets()` para recuperar los ID de consulta para los conjuntos de resultados.

## Parámetros

Nombre	Descripción
<i>IDConsulta</i>	ID de la consulta devuelto desde <code>InitializeQuery()</code> . El texto de SQL para la consulta debe especificar una sentencia CALL.
<i>Resultado</i>	<p>Puntero de una variable VARIANT donde se almacenará el resultado. El resultado es una matriz (tipo de variante VT_ARRAY   VT_VARIANT) que contiene un valor para cada columna del resultado.</p> <p>Cada valor se especifica en su tipo de datos nativo o en el tipo de datos de variante más cercano. Los tipos de valores devueltos con soporte son: serie (tipo de variante VT_BSTR), flotante (tipo de variante VT_R4), doble (tipo de variante VT_R8), corto (tipo de variante VT_I2), largo (tipo de variante VT_I4) y binario (tipo de variante VT_UI1   VT_ARRAY).</p> <p>Deberá inicializar la variable VARIANT correctamente antes de llamar a esta función. Visual Basic efectúa esta acción automáticamente. Los programadores de Visual C++ deben llamar a <code>VariantInit()</code>.</p>
<i>vaConfirmarOK</i>	Valor booleano opcional que especifica si el procedimiento almacenado puede confirmar la unidad de trabajo o si esta operación se debe restringir. El valor por omisión es True.
<i>vaMáxConjuntosResultados</i>	Valor numérico opcional que especifica el número máximo de conjuntos de resultados que se debe permitir que el procedimiento almacenado devuelva. Especifique cero si no desea que el procedimiento almacenado devuelva ningún conjunto de resultados o si el servidor de bases de datos no da soporte a la devolución de conjuntos de resultados por parte de procedimientos almacenados a través de DRDA.
<i>vaNombresColumnas</i>	Valor booleano opcional que especifica si la base de datos debe devolver los nombres de las columnas de cada conjunto de resultados devuelto.
<i>vaEtiquetasColumnas</i>	Valor booleano opcional que especifica si la base de datos debe devolver las etiquetas de las columnas de cada conjunto de resultados devuelto.
<i>vaComentariosColumnas</i>	Valor booleano opcional que especifica si la base de datos debe devolver comentarios para las columnas de cada conjunto de resultados devuelto.

## Valor devuelto

Cero indica que la función se ha completado correctamente; un valor distinto de cero significa que ha habido un error. Si el valor devuelto es distinto de cero, puede llamar a `GetLastErrorString()` o `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` o `GetLastSQLState()` para obtener información adicional del error.

## Export()

`short Export(long IDConsulta, long PrimeraFila, long PrimeraCol, long ÚltimaFila, long ÚltimaCol, short Formato, short DelimitadorSerie, short DelimitadorCol, BOOL IncluirCabecerasCol, BSTR NombreArch, [VARIANT FormatoFechaHora])`

## Descripción

Esta función copia el rango especificado de filas y columnas en el portapapeles. Si no ha recuperado datos de fila para todas las filas que desea copiar en el portapapeles, llame a CompleteQuery() antes de llamar a esta función. Se devolverá un mensaje de error si intenta copiar filas que no se han recuperado de la base de datos.

## Parámetros

Nombre	Descripción
<i>IDConsulta</i>	ID de la consulta devuelto desde InitializeQuery().
<i>PrimeraFila</i>	Primera fila que desea incluir en la exportación.
<i>PrimeraCol</i>	Primera columna que desea incluir en la exportación.
<i>ÚltimaFila</i>	Última fila que desea incluir en la exportación o -1 si se incluyen todas las filas.
<i>ÚltimaCol</i>	Última columna que desea incluir en la exportación o -1 si se incluyen todas las columnas.
<i>IncluirCabecerasCol</i>	Utilice un valor distinto de cero para incluir las cabeceras de columna en la primera fila y cero para no incluirlas.
<i>NombreArch</i>	Serie que contiene el nombre del archivo en el que desea guardar la exportación.
<i>FormatoFechaHora</i>	Opcionalmente, formato a utilizar para los valores de fecha y hora. Los valores válidos son 0 (formato ISO), 1 (formato USA), 2 (formato EUR), 3 (formato JIS) o 4 (formato de panel de control de Windows). El valor por omisión es 4.

**Nota:** El valor de la primera fila en un conjunto de resultados es 0 y el valor de la última fila es el número total de filas menos uno. El valor de la primera columna en un conjunto de resultados es 0 y el valor de la última columna es el número total de columnas menos uno.

Nombre	Descripción
<i>Formato</i>	Especifica el formato de salida.

Valor	Significado
0 (RSEF_TEXT)	El archivo de salida se grabará en formato de sólo texto.
1 (RSEF_HTML)	El archivo de salida se grabará en formato HTML y los datos se organizarán en un tabla HTML.
2 (RSEF_CSV)	El archivo de salida se grabará en formato CSV (valores separados por comas).
3 (RSEF_PCIXF)	El archivo de salida se grabará en formato PC/IXF.
4 (RSEF_S370IXF)	El archivo de salida se grabará en formato IXF de System/370.

Nombre	Descripción
<i>DelimitadorSerie</i>	Especifica el delimitador de serie. Este parámetro se pasa por alto si <i>Formato</i> es RSEF_HTML.

Valor	Significado
-------	-------------

0 (RSSD_NONE)	No se utiliza delimitador de serie.
1 (RSSD_SINGLEQUOTE)	Las series se delimitan mediante comillas simples (').
2 (RSSD_DOUBLEQUOTE)	Las series se delimitan mediante comillas dobles (").

Nombre	Descripción
<i>DelimitadorCol</i>	Especifica el delimitador de columna. Este parámetro se pasa por alto si <i>Formato</i> es RSEF_HTML.

Valor	Significado
0 (RSCD_SPACE)	Las columnas se delimitan mediante un espacio ( ).
1 (RSCD_TAB)	Las columnas se delimitan mediante un tabulador (\t).
2 (RSCD_COMMA)	Las columnas se delimitan mediante una coma (,).

### Valor devuelto

Cero indica que la función se ha completado correctamente; un valor distinto de cero significa que ha habido un error. Si el valor devuelto es distinto de cero, puede llamar a `GetLastErrorString()` o `GetLastErrorType()` para obtener información adicional del error. Si el conjunto de resultados está vacío o no se ha recuperado ninguna fila de la base de datos, se devuelve un valor distinto de cero a no ser que *PrimeraFila*=0 y *ÚltimaFila*=1. En este caso, se devuelve cero y se graba una serie vacía.

### Temas relacionados

`CopyToClipboard()`

## ExportForm()

`short ExportForm(BSTR PropietarioYNombre, BSTR NombreArch)`

### Descripción

Esta función exporta el formato de QMF especificado al archivo indicado.

### Parámetros

Nombre	Descripción
<i>PropietarioYNombre</i>	Serie que contiene el propietario y el nombre, separados por un punto, del objeto que se desea exportar. Por ejemplo, Juan.Consulta2
<i>NombreArch</i>	Serie que contiene el nombre del archivo en el que desea guardar el formato exportado.

### Valor devuelto

Cero indica que la función se ha completado correctamente; un valor distinto de cero significa que ha habido un error. Si el valor devuelto es distinto de cero, puede llamar a `GetLastErrorString()` o `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` o `GetLastSQLState()` para obtener información adicional del error.

### Temas relacionados

`PrintReport()`

## ExportReport()

short ExportReport(long *IDConsulta*, short *TipoOrigen*, BSTR *Origen*, BSTR *NombreArchSalida*, short *LongPágina*, short *AnchoPág*, BOOL *IncluirFechaHora*, BOOL *IncluirNúmPág*, [VARIANT *Formato*], [VARIANT *UsarDefinPágForm*])

### Descripción

Esta función crea un informe para la consulta especificada y lo graba en un archivo. El formato y el diseño del informe deben especificarse en un formato QMF. El archivo de salida es un archivo de texto ASCII, en el que cada línea está separada por un par de caracteres de retorno de carro y salto de línea y cada página está separada por un carácter de salto de página. Es aconsejable visualizar el archivo de salida utilizando una fuente de paso fijo.

### Parámetros

Nombre	Descripción
<i>IDConsulta</i>	ID de la consulta devuelto desde InitializeQuery().
<i>Origen</i>	Nombre (nombre de archivo o Nombre.Proprietario) del formato que desea utilizar.
<i>NombreArchSalida</i>	Nombre del archivo en el que desea grabar el informe.
<i>LongPágina</i>	Número de líneas de cada página del informe. Un parámetro <i>LongPágina</i> de -1 especifica salida continua (sin saltos de página a no ser que el informe tenga un ancho mayor de <i>AnchoPág</i> ).
<i>IncluirFechaHora</i>	Un valor distinto de cero especifica que se incluye la fecha y hora en la parte inferior de cada página. Cero especifica que no se incluye la fecha y hora.
<i>IncluirNúmPág</i>	Un valor distinto de cero especifica que se incluyen números de página en la parte inferior de cada página. Cero especifica que no se incluyen números de página.
<i>Formato</i>	Opcionalmente, especifica el formato del informe exportado. Si es cero, el formato es de sólo texto, especificando que la salida debe ser tal como se genera en el formato (texto o HTML, según el tipo de formato). Si es distinto de cero, el formato es HTML, especificando que la salida debe ser HTML. Con formatos que no son HTML, la salida se convierte a HTML mediante la adición de códigos HTML al principio y al final de la salida. El valor por omisión es cero.
<i>FormatoFechaHora</i>	Opcionalmente, formato a utilizar para los valores de fecha y hora. Los valores válidos son 0 (formato ISO), 1 (formato USA), 2 (formato EUR), 3 (formato JIS) o 4 (formato de panel de control de Windows). El valor por omisión es 4.
<i>Formato</i>	Formato del archivo de salida.
<i>UsarDefinPágForm</i>	Opcionalmente, si el valor es distinto de cero especifica que los parámetros <i>LongPágina</i> , <i>AnchoPág</i> , <i>IncluirFechaHora</i> y <i>IncluirNúmPág</i> se deben ignorar y los valores para los mismos deben tomarse de los valores guardados con el formato especificado. El valor de omisión es cero.
Valor	Significado
0 (RSF_DEFAULT)	Utilice el formato por omisión. <i>NombreForm</i> debe ser una serie vacía.

1 (RSF_DATABASE)	Utilice un formato de la base de datos. Especifique el nombre y el propietario del formato (Nombre.Propietario) en el parámetro <i>NombreForm</i> . Para utilizar un formato ubicado en un servidor de bases de datos diferente, primero utilice <code>ExportForm()</code> para exportar el formato a un archivo y luego especifique un <i>TipoOrigen</i> de <code>RSF_FILE</code> .
2 (RSF_FILE)	Utilice un formato contenido en un archivo. Especifique el nombre del archivo en el parámetro <i>NombreForm</i> .

### Valor devuelto

Cero indica que la función se ha completado correctamente; un valor distinto de cero significa que ha habido un error. Si el valor devuelto es distinto de cero, puede llamar a `GetLastErrorString()` o `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` o `GetLastSQLState()` para obtener información adicional del error.

### Temas relacionados

`ExportForm()`

## FastSaveData()

short `FastSaveData(long IDConsulta, BOOL Sustituir, BSTR NombreTabla, BSTR NombreEspacioTabla, [VARIANT Comentario])`

### Descripción

Esta función crea un informe para la consulta especificada y lo graba en un archivo. El formato y el diseño del informe deben especificarse en un formato QMF. El archivo de salida es un archivo de texto ASCII, en el que cada línea está separada por un par de caracteres de retorno de carro y salto de línea y cada página está separada por un carácter de salto de página. Es aconsejable visualizar el archivo de salida utilizando una fuente de paso fijo.

### Parámetros

Nombre	Descripción
<i>IDConsulta</i>	ID de la consulta devuelto desde <code>InitializeQuery()</code> .
<i>Sustituir</i>	Utilice un valor distinto de cero si desea que los datos especificados sustituyan los datos existentes en la tabla. Utilice cero si desea que los datos especificados se añadan a los datos existentes en la tabla.
<i>NombreTabla</i>	Nombre de la tabla en la que desea almacenar los datos. Si la tabla no existe, QMF para Windows la creará.
<i>NombreEspacioTabla</i>	Nombre del espacio de tabla en el que existe o se creará la tabla. Si <i>NombreEspacioTabla</i> es NULL o una serie vacía, QMF para Windows utilizará el espacio de tabla por omisión. Si ha configurado QMF para Windows para que utilice siempre el espacio de tabla por omisión, se ignorará este parámetro. Consulte <code>RSR_SDDIFFERENTTS</code> en la descripción de <code>GetResourceLimit()</code> .
<i>Comentario</i>	Opcionalmente, una serie que especifica un comentario para la tabla en la que se han guardado los datos.

### Valor devuelto

Cero indica que la función se ha completado correctamente; un valor distinto de cero significa que ha habido un error. Si el valor devuelto es distinto de cero,

puede llamar a `GetLastErrorString()` o `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` o `GetLastSQLState()` para obtener información adicional del error.

## FetchNextRow()

short FetchNextRow(long *IDConsulta*, VARIANT\* *Fila*)

### Descripción

Esta función recupera la fila siguiente de datos de la base de datos.

### Parámetros

Nombre	Descripción
<i>IDConsulta</i>	ID de la consulta devuelto desde <code>InitializeQuery()</code> .
<i>Fila</i>	<p>Puntero de una variable VARIANT donde se almacenará el resultado. El resultado es una matriz (tipo de variante VT_ARRAY   VT_VARIANT) que contiene un valor para cada columna del resultado. Llame a <code>GetColumnCount()</code> para determinar el número de valores de la matriz.</p> <p>Cada valor se especifica en su tipo de datos nativo en el tipo de datos de variante más cercano. Los tipos de valores devueltos con soporte son: serie (tipo de variante VT_BSTR), flotante (tipo de variante VT_R4), doble (tipo de variante VT_R8), corto (tipo de variante VT_I2), largo (tipo de variante VT_I4) y binario (tipo de variante VT_UI1   VT_ARRAY).</p> <p>Una vez que se ha alcanzado el final del conjunto de resultados (no hay más filas a buscar) o si el conjunto de resultados está vacío, el resultado está vacío (tipo de variante VT_EMPTY) en lugar de ser una matriz .</p> <p>Deberá inicializar la variable VARIANT correctamente antes de llamar a esta función. Visual Basic efectúa esta acción automáticamente. Los programadores de Visual C++ deben llamar a <code>VariantInit()</code>.</p>

**Nota:** Debido a un problema en Microsoft Excel 7.0 y Microsoft Access 7.0 (y posiblemente en otros productos Microsoft de 32 bits que utilizan Visual Basic para Aplicaciones), puede que los datos de series de las variables Variant recibidos de QMF para Windows no se conviertan de Unicode (utilizado por OLE) a ANSI (utilizado por VBA). Si ocurre esto, sólo se visualiza el primer carácter de la serie. Para solucionar este problema, establezca la variable igual a una serie vacía antes de llamar a la función QMF para Windows que utiliza la variable.

### Valor devuelto

Cero indica que la función se ha completado correctamente; un valor distinto de cero significa que ha habido un error. Cuando se alcanza el final del conjunto de resultados, el valor devuelto es -1. Si el valor devuelto es distinto de cero, puede llamar a `GetLastErrorString()` o `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` o `GetLastSQLState()` para obtener información adicional del error.

### Temas relacionados

`FetchNextRows()`

## FetchNextRowEx()

short FetchNextRowEx(long *IDConsulta*)

### Descripción

Esta función recupera la fila siguiente de datos de la base de datos. Puede utilizar esta función en entornos que no soportan matrices VARIANT, por ejemplo Microsoft Access 2.0. Utilice esta función junto con GetColumnValue() para recuperar los datos de cada columna de la fila actual.

### Parámetros

Nombre	Descripción
<i>IDConsulta</i>	ID de la consulta devuelto desde InitializeQuery().

### Valor devuelto

Cero indica que la función se ha completado correctamente; un valor distinto de cero significa que ha habido un error. Cuando se alcanza el final del conjunto de resultados, el valor devuelto es -1. Si el valor devuelto es distinto de cero, puede llamar a GetLastErrorString() o GetLastErrorType(), GetLastSQLCode(), GetLastSQLError() o GetLastSQLState() para obtener información adicional del error.

### Temas relacionados

FetchNextRowsEx()

## FetchNextRows()

short FetchNextRows(long *IDConsulta*, VARIANT\* *Filas*, long\* *NúmFilas*)

### Descripción

Esta función recupera las siguientes filas de datos *NúmFilas* de la base de datos.

### Parámetros

Nombre	Descripción
<i>IDConsulta</i>	ID de la consulta devuelto desde InitializeQuery().

Fila	<p>Puntero de una variable VARIANT donde se almacenará el resultado. El resultado es una matriz de dos dimensiones (tipo de variante VT_ARRAY   VT_VARIANT) que contiene un valor para cada columna de cada fila. Llame a GetColumnCount() para determinar el número de columnas de la matriz. Las dimensiones de la matriz son [NúmFilas][NúmCol], incluso si el número de filas no recuperadas en el conjunto de resultados es inferior a NúmFilas (en este caso, la matriz contendrá entradas adicionales no utilizadas).</p> <p>Cada valor se especifica en su tipo de datos nativo en el tipo de datos de variante más cercano. Los tipos de valores devueltos con soporte son: serie (tipo de variante VT_BSTR), flotante (tipo de variante VT_R4), doble (tipo de variante VT_R8), corto (tipo de variante VT_I2), largo (tipo de variante VT_I4) y binario (tipo de variante VT_UI1   VT_ARRAY).</p> <p>Una vez que se ha alcanzado el final del conjunto de resultados (no hay más filas a buscar) o si el conjunto de resultados está vacío, el resultado está vacío (tipo de variante VT_EMPTY) en lugar de ser una matriz.</p> <p>Deberá inicializar la variable VARIANT correctamente antes de llamar a esta función. Visual Basic efectúa esta acción automáticamente. Los programadores de Visual C++ deben llamar a VariantInit().</p>
NúmFilas	<p>Un puntero es un valor largo que contiene el número de filas que se debe recuperar. Si el número de filas no recuperadas del conjunto de resultados es menor que NúmFilas, NúmFilas se restablece en el número real de filas contenidas en el resultado.</p>

**Nota:** Debido a un problema en Microsoft Excel 7.0 y Microsoft Access 7.0 (y posiblemente en otros productos Microsoft de 32 bits que utilizan Visual Basic para Aplicaciones), puede que los datos de series de las variables Variant recibidos de QMF para Windows no se conviertan de Unicode (utilizado por OLE) a ANSI (utilizado por VBA). Si ocurre esto, sólo se visualiza el primer carácter de la serie. Para solucionar este problema, establezca la variable igual a una serie vacía antes de llamar a la función QMF para Windows que utiliza la variable.

### Valor devuelto

Cero indica que la función se ha completado correctamente; un valor distinto de cero significa que ha habido un error. Cuando se alcanza el final del conjunto de resultados, el valor devuelto es -1. Si el valor devuelto es distinto de cero, puede llamar a GetLastErrorString() o GetLastErrorType(), GetLastSQLCode(), GetLastSQLError() o GetLastSQLState() para obtener información adicional del error.

### Temas relacionados

FetchNextRow()

## FetchNextRowsEx()

short FetchNextRowsEx(long IDConsulta, long\* NúmFilas)

### Descripción

Esta función recupera las siguientes filas de datos NúmFilas de la base de datos. Puede utilizar esta función en entornos que no soportan matrices VARIANT, por

ejemplo Microsoft Access 2.0. Utilice esta función junto con `GetColumnValueEx()` para recuperar los datos de cada columna de una fila determinada.

### Parámetros

Nombre	Descripción
<i>IDConsulta</i>	ID de la consulta devuelto desde <code>InitializeQuery()</code> .
NúmFilas	Un puntero es un valor largo que contiene el número de filas que se debe recuperar. Si el número de filas no recuperadas del conjunto de resultados es menor que <i>NúmFilas</i> , <i>NúmFilas</i> se restablece en el número real de filas contenidas en el resultado.

### Valor devuelto

Cero indica que la función se ha completado correctamente; un valor distinto de cero significa que ha habido un error. Cuando se alcanza el final del conjunto de resultados, el valor devuelto es -1. Si el valor devuelto es distinto de cero, puede llamar a `GetLastErrorString()` o `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` o `GetLastSQLState()` para obtener información adicional del error.

### Temas relacionados

`FetchNextRowEx()`

## FlushQMFCache()

`void FlushQMFCache()`

### Descripción

Esta función indica a QMF para Windows que debe vaciar su antememoria de información de QMF, eliminando su contenido. La próxima vez que QMF para Windows necesite información de QMF, la obtendrá de la base de datos. Normalmente QMF para Windows pone la información de QMF procedente de la base de datos en la antememoria para reducir así el tráfico de la base de datos y mejorar el rendimiento. Llame a esta función antes de llamar a `GetQMFObjectInfo()`, `GetQMFQueryText()`, o `GetQMFObjectList()` para asegurarse de que la información devuelta por estas funciones está actualizada.

### Valor devuelto

Ninguno.

## GetColumnCount()

`long GetColumnCount(long IDConsulta)`

### Descripción

Esta función devuelve el número de columnas del conjunto de resultados.

### Parámetros

Nombre	Descripción
<i>IDConsulta</i>	ID de la consulta devuelto desde <code>InitializeQuery()</code> .

### Valor devuelto

Número de columnas de cada fila, si la función es satisfactoria. Si no es satisfactoria, 0 ó -1. Si el valor devuelto es 0 ó -1, puede llamar a `GetLastErrorString()` o `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` o `GetLastSQLState()` para obtener información adicional del error.

## GetColumnDataValue()

short GetColumnDataValue(long *IDConsulta*, long *Índice*)

### Descripción

Esta función devuelve el valor de datos para la columna especificada en *Índice* para la fila actual de datos. Después de llamar a esta función, se puede interrogar a la propiedad *Valor* por el error devuelto. Utilice esta función con FetchNextRowEx() para acceder a los datos en una sola fila de datos.

### Parámetros

Nombre	Descripción
<i>IDConsulta</i>	ID de la consulta devuelto desde InitializeQuery().
<i>Índice</i>	El índice con base cero de la fila de datos que desea recuperar.

### Valor devuelto

Cero indica que la función se ha completado correctamente; un valor distinto de cero significa que ha habido un error. Si el valor devuelto es distinto de cero, puede llamar a GetLastErrorString() o GetLastErrorType(), GetLastErrorSQLCode(), GetLastErrorSQLError() o GetLastErrorSQLState() para obtener información adicional del error.

## GetColumnHeader()

BSTR GetColumnHeader(long *IDConsulta*, long *Índice*, short\* *Resultado*)

### Descripción

Esta función devuelve el encabezamiento de columna (nombre de columna) asociada con el índice *Índice*.

### Parámetros

Nombre	Descripción
<i>IDConsulta</i>	ID de la consulta devuelto desde InitializeQuery().
<i>Índice</i>	El índice con base cero de la fila de datos que desea recuperar.
<i>Resultado</i>	Cero indica que la función se ha completado correctamente; un valor distinto de cero significa que ha habido un error. Si el valor devuelto es distinto de cero, puede llamar a GetLastErrorString() o GetLastErrorType(), GetLastErrorSQLCode(), GetLastErrorSQLError() o GetLastErrorSQLState() para obtener información adicional del error.

**Nota:** Las cabeceras de columna no están disponibles para sentencias de de SQL estático. Para los ID de consulta devueltos desde InitializeStaticQuery(), GetColumnHeader devuelve una serie del formato "Coln" donde "n" es el número de columna.

### Valor devuelto

La serie devuelta representa el nombre de columna especificado en el parámetro *Índice*.

## GetColumnHeaderEx()

short GetColumnHeaderEx(long *IDConsulta*, long *Índice*)

## Descripción

Esta función devuelve el encabezamiento de columna (nombre de columna) asociada con el índice *Índice*. Después de llamar a esta función, se puede interrogar a la propiedad *Valor* por el error devuelto.

## Parámetros

Nombre	Descripción
<i>IDConsulta</i>	ID de la consulta devuelto desde <code>InitializeQuery()</code> .
Índice	El índice con base cero de la fila de datos que desea recuperar.

**Nota:** Las cabeceras de columna no están disponibles para sentencias de de SQL estático. Para los ID de consulta devueltos desde `InitializeStaticQuery()`, `GetColumnHeader` devuelve una serie del formato "Coln" donde "n" es el número de columna.

## Valor devuelto

Cero indica que la función se ha completado correctamente; un valor distinto de cero significa que ha habido un error. Si el valor devuelto es cero, consulte la propiedad *Valor* para obtener la serie que representa el nombre de columna. Si el valor devuelto es distinto de cero, puede llamar a `GetLastErrorString()` o `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` o `GetLastSQLState()` para obtener información adicional del error.

## GetColumnHeadings()

short `GetColumnHeadings(long IDConsulta, VARIANT* Cabeceras)`

## Descripción

Esta función devuelve las cabeceras de columna (también denominadas nombres de columnas).

## Parámetros

Nombre	Descripción
<i>IDConsulta</i>	ID de la consulta devuelto desde <code>InitializeQuery()</code> .
Cabeceras	Puntero de una variable <code>VARIANT</code> donde se almacenará el resultado. El resultado es una matriz de series (tipo de variante <code>VT_ARRAY   VT_BSTR</code> ) que contiene una serie para cada cabecera de columna.  Deberá inicializar la variable <code>VARIANT</code> correctamente antes de llamar a esta función. Visual Basic efectúa esta acción automáticamente. Los programadores de Visual C++ deben llamar a <code>VariantInit()</code> .

**Nota:** Debido a un problema en Microsoft Excel 7.0 y Microsoft Access 7.0 (y posiblemente en otros productos Microsoft de 32 bits que utilizan Visual Basic para Aplicaciones), puede que los datos de series de las variables `Variant` recibidos de QMF para Windows no se conviertan de Unicode (utilizado por OLE) a ANSI (utilizado por VBA). Si ocurre esto, sólo se visualiza el primer carácter de la serie. Para solucionar este problema, establezca la variable igual a una serie vacía antes de llamar a la función QMF para Windows que utiliza la variable.

**Nota:** Las cabeceras de columna no están disponibles para sentencias de SQL estático. Para los ID de consulta devueltos desde `InitializeStaticQuery()`, `GetColumnHeadings` devolverá las series "Col1", "Col2", etc.

### Valor devuelto

Cero indica que la función se ha completado correctamente; un valor distinto de cero significa que ha habido un error. Si el valor devuelto es distinto de cero, puede llamar a `GetLastErrorString()` o `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` o `GetLastSQLState()` para obtener información adicional del error.

## GetColumnValue()

short `GetColumnValue(long IDConsulta, long Índice, VARIANT* Valor)`

### Descripción

Esta función devuelve el valor de datos para la columna especificada en *Índice* para la fila actual de datos. Utilice esta función con `FetchNextRowEx()` para acceder a los datos en una sola fila de datos.

### Parámetros

Nombre	Descripción
<i>IDConsulta</i>	ID de la consulta devuelto desde <code>InitializeQuery()</code> .
Índice	El índice con base cero de la fila de datos que desea recuperar.
Valor	Puntero de una variable VARIANT donde desea almacenar los resultados. El resultado es un valor de datos basado en el tipo de variante.  Deberá inicializar la variable VARIANT correctamente antes de llamar a esta función. Visual Basic efectúa esta acción automáticamente. Los programadores de Visual C++ deben llamar a <code>VariantInit()</code> .

### Valor devuelto

Cero indica que la función se ha completado correctamente; un valor distinto de cero significa que ha habido un error. Si el valor devuelto es distinto de cero, puede llamar a `GetLastErrorString()` o `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` o `GetLastSQLState()` para obtener información adicional del error.

## GetColumnValueEx()

short `GetColumnValueEx(long IDConsulta, long ÍndiceFila, long ÍndiceCol, VARIANT* Valor)`

### Descripción

Esta función devuelve el valor de datos para la columna especificada en *ÍndiceCol* para la fila de datos especificada en *ÍndiceFila*. Utilice esta función con `FetchNextRowsEx()` para acceder a los datos de una sola fila de datos.

### Parámetros

Nombre	Descripción
<i>IDConsulta</i>	ID de la consulta devuelto desde <code>InitializeQuery()</code> .
ÍndiceFila	El índice con base cero de la fila que desea recuperar.

ÍndiceCol	El índice con base cero de la columna que desea recuperar.
Valor	<p>Puntero de una variable VARIANT donde desea almacenar el resultado. Puede consultar la variante resultante para averiguar el tipo de datos para procesos posteriores.</p> <p>Deberá inicializar la variable VARIANT correctamente antes de llamar a esta función. Visual Basic efectúa esta acción automáticamente. Los programadores de Visual C++ deben llamar a VariantInit().</p>

### Valor devuelto

Cero indica que la función se ha completado correctamente; un valor distinto de cero significa que ha habido un error. Si el valor devuelto es distinto de cero, puede llamar a GetLastErrorString() o GetLastErrorType(), GetLastErrorSQLCode(), GetLastErrorSQLError() o GetLastErrorSQLState() para obtener información adicional del error.

## GetDefaultServerName()

BSTR GetDefaultServerName()

### Descripción

Esta función devuelve una serie que contiene el nombre del servidor por omisión.

### Valor devuelto

Serie que especifica el nombre del servidor por omisión.

## GetGlobalVariable()

BSTR GetGlobalVariable(BSTR *Nombre*)

### Descripción

Esta función recupera el valor de la variable global especificada.

### Parámetros

Nombre	Descripción
Nombre	Serie que contiene el nombre de la variable que desea establecer.

### Valor devuelto

Serie que contiene el valor de la variable global o NULL si la variable no tiene ningún valor o se produce un error.

## GetHostVariableNames()

short GetHostVariableNames(long *IDConsulta*, VARIANT\* *Nombres*)

### Descripción

Esta función devuelve una matriz de los nombres de todas las variables del sistema principal a las que se hace referencia en la consulta especificada. La consulta debe ser una consulta estática que haga referencia a las variables del sistema principal (o bien almacenadas con la consulta QMF o creadas mediante AddHostVariable()).

### Parámetros

Nombre	Descripción
--------	-------------

<i>IDConsulta</i>	ID de la consulta devuelto desde InitializeQuery().
Nombres	Puntero de una variable VARIANT donde desea almacenar la matriz de resultados.

### Valor devuelto

Cero indica que la función se ha completado correctamente; un valor distinto de cero significa que ha habido un error. Si el valor devuelto es distinto de cero, puede llamar a GetLastErrorString() para obtener información adicional del error.

## GetHostVariableTypes()

short GetHostVariableTypes(long *IDConsulta*, VARIANT\* *Tipos*)

### Descripción

Esta función devuelve una matriz de los tipos de datos de todas las variables del sistema principal a las que se hace referencia en la consulta especificada. La consulta debe ser una consulta estática que haga referencia a las variables del sistema principal (o bien almacenadas con la consulta QMF o creadas mediante AddHostVariable()). Consulte AddHostVariable() para obtener una lista de tipos de datos que se pueden devolver.

### Parámetros

Nombre	Descripción
<i>IDConsulta</i>	ID de la consulta devuelto desde InitializeQuery().
Tipos	Puntero de una variable VARIANT donde desea almacenar la matriz de resultados.

### Valor devuelto

Cero indica que la función se ha completado correctamente; un valor distinto de cero significa que ha habido un error. Si el valor devuelto es distinto de cero, puede llamar a GetLastErrorString() para obtener información adicional del error.

## GetLastErrorString()

BSTR GetLastErrorString()

### Descripción

Esta función devuelve una serie que contiene información sobre el error más reciente. Si llama a esta función después de una función que se ha ejecutado satisfactoriamente (sin errores), esta función devuelve información acerca del último error producido durante una llamada de función anterior. Para evitar confusiones, llame siempre a esta función inmediatamente después de llamar a una función que ha devuelto un error.

### Valor devuelto

Una serie que contiene información sobre errores. Si no se ha producido ningún error desde que se creó el objeto de la API de QMF, se devolverá NULL.

### Temas relacionados

GetLastErrorType()  
 GetLastErrorSQLCode()  
 GetLastErrorSQLError()  
 GetLastErrorSQLState()

## GetLastErrorType()

short GetLastErrorType()

### Descripción

Esta función devuelve el tipo de error más reciente. Si llama a esta función después de una función que se ha ejecutado satisfactoriamente (sin errores), esta función devuelve información acerca del último error producido durante una llamada de función anterior. Para evitar confusiones, llame siempre a esta función inmediatamente después de llamar a una función que ha devuelto un error.

### Valor devuelto

El valor devuelto indica el tipo de error:

Valor	Significado
0 (RS_ERROR_NONE)	No se ha producido ningún error desde la creación del objeto de la API de QMF para Windows.
1 (RS_ERROR_SQL)	Se ha producido un error de SQL. Si el error se ha producido durante una llamada a una función que toma <i>IDConsulta</i> como argumento, llame a <i>Close()</i> para cerrar dicha consulta. No se ha efectuado ninguna retrotracción. Puede continuar utilizando el objeto de la API de QMF para Windows, aunque es posible que encuentre errores adicionales.
2 (RS_ERROR_USER_CANCEL)	Un usuario ha cancelado la operación, generalmente pulsando en Cancelar en la ventana del proceso en curso. Esto hace que QMF para Windows realice una retrotracción implícita (invalidando todos los ID de consulta pendientes) y se interrumpa la conexión con la base de datos. Deberá llamar a <i>InitializeServer()</i> o <i>ReinitializeServer()</i> si desea continuar.
3 (RS_ERROR_FATAL_GOV)	Se ha producido un error de control grave. Una posibilidad es que haya excedido el tiempo de espera de la API de QMF para Windows porque se ha excedido el tiempo de inactividad máximo permitido. Esto hace que QMF para Windows realice una retrotracción implícita (invalidando todos los ID de consulta pendientes) y se interrumpa la conexión con la base de datos. Deberá llamar a <i>InitializeServer()</i> o <i>ReinitializeServer()</i> si desea continuar.
4 (RS_ERROR_NONFATAL_GOV)	Se ha producido un error de control que no es grave. Se ha excedido el número máximo permitido de filas a buscar o el verbo de SQL no está permitido. Si el error se ha producido durante una llamada a una función que toma <i>IDConsulta</i> como argumento, llame a <i>Close()</i> para cerrar dicha consulta. No se realizará ninguna retrotracción y la conexión con la base de datos no se verá afectada, con lo cual podrá seguir utilizando el objeto de la API de QMF para Windows.
5 (RS_ERROR_OTHER)	Se ha producido un error general. No se ha efectuado ninguna retrotracción. Puede continuar utilizando el objeto de la API de QMF para Windows, aunque es posible que encuentre errores adicionales.

## Temas relacionados

GetLastErrorString()  
GetLastSQLCode()  
GetLastSQLError()  
GetLastSQLState()

## GetLastSQLCode()

long GetLastSQLCode()

### Descripción

Esta función devuelve el código de SQL del error más reciente. Si llama a esta función después de una función que se ha ejecutado satisfactoriamente (sin errores), esta función devuelve información acerca del último error producido durante una llamada de función anterior. Para evitar confusiones, llame siempre a esta función inmediatamente después de llamar a una función que ha devuelto un error.

### Valor devuelto

El código de SQL del error más reciente. Si no se ha producido ningún error desde que se creó el objeto de la API de QMF para Windows o el error más reciente no era un error de SQL, se devolverá cero.

## Temas relacionados

GetLastErrorString()  
GetLastErrorType()  
GetLastSQLError()  
GetLastSQLState()

## GetLastSQLError()

VARIANT GetLastSQLError()

### Descripción

Esta función devuelve información detallada sobre el error de SQL más reciente. Si llama a esta función después de una función que se ha ejecutado satisfactoriamente (sin errores), esta función devuelve información acerca del último error producido durante una llamada de función anterior. Para evitar confusiones, llame siempre a esta función inmediatamente después de llamar a una función que ha devuelto un error.

### Valor devuelto

Una matriz (tipo de variante VT\_ARRAY | VT\_VARIANT) que contiene información sobre errores. Si no se ha producido ningún error desde que se creó el objeto de la API de QMF para Windows o el error más reciente no era un error de SQL, se devolverá un valor vacío (tipo de variante VT\_EMPTY). La matriz tiene el formato siguiente:

Elemento	Tipo	Contenido
0	largo (VT_I4)	Código
1	serie (VT_BSTR)	Estado
2	serie (VT_BSTR)	ErrProc
3	serie (VT_BSTR)	RDBName
4	largo (VT_I4)	ErrD1
5	largo (VT_I4)	ErrD2
6	largo (VT_I4)	ErrD3

7	largo (VT_I4)	ErrD4
8	largo (VT_I4)	ErrD5
9	largo (VT_I4)	ErrD6
10	serie (VT_BSTR)	Warn0
11	serie (VT_BSTR)	Warn1
12	serie (VT_BSTR)	Warn2
13	serie (VT_BSTR)	Warn3
14	serie (VT_BSTR)	Warn4
15	serie (VT_BSTR)	Warn5
16	serie (VT_BSTR)	Warn6
17	serie (VT_BSTR)	Warn7
18	serie (VT_BSTR)	Warn8
19	serie (VT_BSTR)	Warn9
20	serie (VT_BSTR)	WarnA
21	serie (VT_BSTR)	MessageTokens

### Temas relacionados

[GetLastErrorString\(\)](#)  
[GetLastErrorType\(\)](#)  
[GetLastSQLCode\(\)](#)  
[GetLastSQLState\(\)](#)

## GetLastSQLState()

BSTR GetLastErrorState()

### Descripción

Esta función devuelve el estado de SQL del error más reciente. Si llama a esta función después de una función que se ha ejecutado satisfactoriamente (sin errores), esta función devuelve información acerca del último error producido durante una llamada de función anterior. Para evitar confusiones, llame siempre a esta función inmediatamente después de llamar a una función que ha devuelto un error.

### Valor devuelto

Una serie que devuelve el código de SQL del error más reciente. Si no se ha producido ningún error desde que se creó el objeto de la API de QMF para Windows o el error más reciente no era un error de SQL, se devolverá NULL.

### Temas relacionados

[GetLastErrorString\(\)](#)  
[GetLastErrorType\(\)](#)  
[GetLastSQLCode\(\)](#)  
[GetLastSQLError\(\)](#)

## GetOption()

short GetOption(short *Opción*, VARIANT\* *Valor*)

### Descripción

Obtiene el valor de la opción especificado en QMF para Windows.

## Parámetros

Nombre	Descripción
<i>Opción</i>	Especifica qué opción se debe recuperar.
Valor	Significado
0 (RSO_SERVER_DEFINITION_FILE)	Nombre de archivo de definición de servidor.
1 (RSO_CPIC_DLL)	Nombre de archivo de la DLL del proveedor de CPI-C.
2 (RSO_CPIC_TIMEOUT_WARNING)	Tiempo de espera de aviso de CPI-C (en segundos). Este límite no se utiliza para la API de QMF para Windows.
3 (RSO_CPIC_TIMEOUT_CANCEL)	Tiempo de espera de cancelación de CPI-C (en segundos).
4 (RSO_TCP_TIMEOUT_WARNING)	Tiempo de espera de aviso de TCP (en segundos). Este límite no se utiliza para la API de QMF para Windows.
5 (RSO_TCP_TIMEOUT_CANCEL)	Tiempo de espera de cancelación de TCP (en segundos).
6 (RSO_DISPLAY_NULLS_STRING)	Serie que se va a utilizar para visualizar valores nulos.
7 (RSO_ENTER_NULLS_STRING)	Serie que se va a utilizar para entrar valores nulos.
8 (RSO_ENTER_DEFAULTS_STRING)	Serie que se va a utilizar para entrar valores por omisión.
9 (RSO_TRACE_FILE_1)	Nombre del archivo de rastreo 1.
10 (RSO_TRACE_FILE_2)	Nombre del archivo de rastreo 2.
11 (RSO_TCP_TRACE_LEVEL)	Nivel de rastreo de TCP.
12 (RSO_CPIC_TRACE_LEVEL)	Nivel de rastreo de CPI-C.
13 (RSO_DDM_TRACE_LEVEL)	Nivel de rastreo de DDM.
Valor	Puntero de una variable VARIANT donde se almacenará el resultado. El resultado es una matriz (tipo de variante VT_ARRAY   VT_VARIANT) que contiene un valor para cada columna del resultado. Llame a GetColumnCount() para determinar el número de valores de la matriz. Deberá inicializar la variable VARIANT correctamente antes de llamar a esta función. Visual Basic efectúa esta acción automáticamente. Los programadores de Visual C++ deben llamar a VariantInit().

**Nota:** Debido a un problema en Microsoft Excel 7.0 y Microsoft Access 7.0 (y posiblemente en otros productos Microsoft de 32 bits que utilizan Visual Basic para Aplicaciones), puede que los datos de series de las variables Variant recibidos de QMF para Windows no se conviertan de Unicode (utilizado por OLE) a ANSI (utilizado por VBA). Si ocurre esto, sólo se visualiza el primer carácter de la serie. Para solucionar este problema, establezca la variable igual a una serie vacía antes de llamar a la función QMF para Windows que utiliza la variable.

### Valor devuelto

Cero indica que la función se ha completado correctamente; un valor distinto de cero significa que ha habido un error. Si el valor devuelto es distinto de cero, puede llamar a `GetLastErrorString()` o `GetLastErrorType()` para obtener información adicional del error.

### Temas relacionados

`SetOption()`

## GetOptionEx()

`short GetOptionEx(short Opción)`

### Descripción

Obtiene el valor de la opción especificado en QMF para Windows. Cuando se devuelva el valor de la opción, deberá consultar la propiedad *Opción* para averiguar los datos.

### Parámetros

Nombre	Descripción
Opción	Los valores de opción son los mismos que los de la llamada a <code>GetOption()</code> .

### Valor devuelto

Cero indica que la función se ha completado correctamente; un valor distinto de cero significa que ha habido un error. Si el valor devuelto es distinto de cero, puede llamar a `GetLastErrorString()` o `GetLastErrorType()` para obtener información adicional del error.

### Temas relacionados

`GetOption()`

`SetOption()`

## GetProcText()

`BSTR GetProcText(long IDProc)`

### Descripción

Esta función devuelve el texto que se ejecuta para el procedimiento especificado, después de la sustitución de variables. Debe utilizar `SetProcVariable()` para establecer el valor de las variables utilizadas en el procedimiento antes de llamar a esta función.

### Parámetros

Nombre	Descripción
IDProc	ID de procedimiento devuelto desde <code>InitializeProc()</code> .

### Valor devuelto

Si no se producen errores, se devuelve una serie que contiene el texto del procedimiento. Si se producen errores, se devuelve `NULL`. Si el valor devuelto es `NULL`, puede llamar a `GetLastErrorString()` o `GetLastErrorType()` para obtener información adicional del error.

## GetProcVariables()

`short GetProcVariables(long IDProc, VARIANT* Valor)`

## Descripción

Obtiene el valor de la opción especificado en QMF para Windows.

## Parámetros

Nombre	Descripción
IDProc	ID de procedimiento devuelto desde InitializeProc().
Valor	Puntero de una variable VARIANT donde se almacenará el resultado. El resultado es una matriz de series (tipo de variante VT_ARRAY   VT_BSTR), cada una de las cuales contiene el nombre de una variable. Si no hay variables en el procedimiento, el resultado estará vacío (tipo de variante VT_EMPTY). Deberá inicializar la variable VARIANT correctamente antes de llamar a esta función. Visual Basic efectúa esta acción automáticamente. Los programadores de Visual C++ deben llamar a VariantInit().

**Nota:** Debido a un problema en Microsoft Excel 7.0 y Microsoft Access 7.0 (y posiblemente en otros productos Microsoft de 32 bits que utilizan Visual Basic para Aplicaciones), puede que los datos de series de las variables Variant recibidos de QMF para Windows no se conviertan de Unicode (utilizado por OLE) a ANSI (utilizado por VBA). Si ocurre esto, sólo se visualiza el primer carácter de la serie. Para solucionar este problema, establezca la variable igual a una serie vacía antes de llamar a la función QMF para Windows que utiliza la variable.

## Valor devuelto

Cero indica que la función se ha completado correctamente; un valor distinto de cero significa que ha habido un error. Si no hay variables en el procedimiento, el valor devuelto será RS\_NO\_ERROR\_NO\_DATA (-1). Si el valor devuelto es distinto de cero, puede llamar a GetLastErrorString() o GetLastErrorType() para obtener información adicional del error.

## GetQMFObjectInfo()

short GetQMFObjectInfo(BSTR PropietarioYNombre, short Tipo, short Hora, VARIANT\* Valor)

## Descripción

Esta función devuelve información sobre un objeto de QMF (un formato o una consulta). Los parámetros *Tipo* y *Hora* especifican la información devuelta.

## Parámetros

Nombre	Descripción
PropietarioYNombre	Serie que contiene el propietario y el nombre, separados por un punto, del objeto para el que se desea recuperar información. Por ejemplo, Juan.Consulta2
Valor	Puntero de una variable VARIANT donde se almacenará el resultado. Para RSI_TIMEUSED, RSI_TIMESRUN, RSI_TIMESCANCELLED y RSI_LEVEL, el resultado es un valor corto (tipo de variante VT_I2). Para RSI_RESTRICTED el resultado es un valor booleano (tipo de variante VT_BOOL). Para el resto, el resultado es una serie (tipo de variante VT_BSTR). Deberá inicializar la variable VARIANT correctamente antes de llamar a esta función. Visual Basic efectúa esta acción automáticamente. Los programadores de Visual C++ deben llamar a VariantInit().

**Nota:** Debido a un problema en Microsoft Excel 7.0 y Microsoft Access 7.0 (y posiblemente en otros productos Microsoft de 32 bits que utilizan Visual Basic para Aplicaciones), puede que los datos de series de las variables Variant recibidos de QMF para Windows no se conviertan de Unicode (utilizado por OLE) a ANSI (utilizado por VBA). Si ocurre esto, sólo se visualiza el primer carácter de la serie. Para solucionar este problema, establezca la variable igual a una serie vacía antes de llamar a la función QMF para Windows que utiliza la variable.

<i>Tipo</i>	Especifica el tipo de información que se debe obtener.
Valor	Significado
0 (RSI_COMMENT)	Comentario
1 (RSI_LEVEL)	Nivel
2 (RSI_TYPE)	Tipo
3 (RSI_SUBTYPE)	Subtipo
4 (RSI_RESTRICTED)	Restringido
5 (RSI_MODEL)	Modelo
6 (RSI_TIMESUSED)	Número de veces utilizado.
7 (RSI_TIMESRUN)	Número de veces ejecutado.
8 (RSI_TIMESCANCELLED)	Número de veces cancelado.
9 (RSI_DATE)	Fecha de la primera o última vez que se ha utilizado o fecha de la última vez que se ha modificado.
10 (RSI_TIME)	Hora de la primera o última vez que se ha utilizado u hora de la última vez que se ha modificado.
11 (RSI_USERID)	ID de usuario de la primera o última vez que se ha utilizado o ID de usuario de la última vez que se ha modificado.
12 (RSI_SQLID)	ID de SQL de la primera o última vez que se ha utilizado o ID de SQL de la última vez que se ha modificado.
13 (RSI_ENVIRONMENT)	Entorno de la primera o última vez que se ha utilizado o entorno de la última vez que se ha modificado.
14 (RSI_MODE)	Modalidad de la primera o última vez que se ha utilizado o modalidad de la última vez que se ha modificado.
15 (RSI_COMMAND)	Mandato de la primera o última vez que se ha utilizado o mandato de la última vez que se ha modificado.
<i>Hora</i>	Especifica la primera o la última vez que se ha utilizado o la última vez que se ha modificado.
Valor	Significado

0 (RST_FIRSTUSED)	Primera utilización.
1 (RST_LASTUSED)	Última utilización.
2 (RST_LASTMODIFIED)	Última modificación.

### Valor devuelto

Cero indica que la función se ha completado correctamente; un valor distinto de cero significa que ha habido un error. Si el valor devuelto es distinto de cero, puede llamar a `GetLastErrorString()` o `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` o `GetLastSQLState()` para obtener información adicional del error.

## GetQMFOBJECTINFOEX()

short GetQMFOBJECTINFOEX(BSTR PropietarioYNombre, short Tipo, short Hora)

### Descripción

Esta función devuelve información sobre el objeto de QMF. Los parámetros *Tipo* y *Hora* especifican la información devuelta. Después de llamar a esta función, se puede interrogar a la propiedad *InfoObjQMF* por el valor devuelto.

### Parámetros

Nombre	Descripción
PropietarioYNombre	Serie que contiene el propietario y el nombre, separados por un punto, del objeto para el que se desea recuperar información. Por ejemplo, Juan.Consulta2
Tipo	Especifica el tipo de información que se debe obtener.

Valor	Significado
0 (RSI_COMMENT)	Comentario
1 (RSI_LEVEL)	Nivel
2 (RSI_TYPE)	Tipo
3 (RSI_SUBTYPE)	Subtipo
4 (RSI_RESTRICTED)	Restringido
5 (RSI_MODEL)	Modelo
6 (RSI_TIMESUSED)	Número de veces utilizado.
7 (RSI_TIMESRUN)	Número de veces ejecutado.
8 (RSI_TIMESCANCELLED)	Número de veces cancelado.
9 (RSI_DATE)	Fecha de la primera o última vez que se ha utilizado o fecha de la última vez que se ha modificado.
10 (RSI_TIME)	Hora de la primera o última vez que se ha utilizado u hora de la última vez que se ha modificado.
11 (RSI_USERID)	ID de usuario de la primera o última vez que se ha utilizado o ID de usuario de la última vez que se ha modificado.

12 (RSI_SQLID)	ID de SQL de la primera o última vez que se ha utilizado o ID de SQL de la última vez que se ha modificado.
13 (RSI_ENVIRONMENT)	Entorno de la primera o última vez que se ha utilizado o entorno de la última vez que se ha modificado.
14 (RSI_MODE)	Modalidad de la primera o última vez que se ha utilizado o modalidad de la última vez que se ha modificado.
15 (RSI_COMMAND)	Mandato de la primera o última vez que se ha utilizado o mandato de la última vez que se ha modificado.
<i>Hora</i>	Especifica la primera o la última vez que se ha utilizado o la última vez que se ha modificado.
<b>Valor</b>	<b>Significado</b>
0 (RST_FIRSTUSED)	Primera utilización.
1 (RST_LASTUSED)	Última utilización.
2 (RST_LASTMODIFIED)	Última modificación.

### Valor devuelto

Cero indica que la función se ha completado correctamente; un valor distinto de cero significa que ha habido un error. Si el valor devuelto es distinto de cero, puede llamar a `GetLastErrorString()` o `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` o `GetLastSQLState()` para obtener información adicional del error.

## GetQMFOBJECTLIST()

short GetQMFOBJECTLIST(BSTR *Propietario*, BSTR *Nombre*, short *Tipo*, VARIANT\* *Lista*)

### Descripción

Esta función devuelve una matriz de los nombres de todos los objetos de QMF que coinciden con los patrones especificados en los parámetros *Propietario* y *Nombre*.

### Parámetros

Nombre	Descripción
Propietario	Serie que contiene el propietario de los objetos que desea incluir en la lista devuelta.
Nombre	Serie que contiene el nombre de los objetos que desea incluir en la lista devuelta.
Lista	Puntero de una variable VARIANT donde se almacenará el resultado. El resultado es una matriz de series (tipo de variante VT_ARRAY   VT_BSTR), cada una de las cuales tiene el formato Propietario.Nombre. Si no se encuentran consultas de QMF para Windows coincidentes, el resultado estará vacío (tipo de variante VT_EMPTY). Deberá inicializar la variable VARIANT correctamente antes de llamar a esta función. Visual Basic efectúa esta acción automáticamente. Los programadores de Visual C++ deben llamar a <code>VariantInit()</code> .

**Nota:** Debido a un problema en Microsoft Excel 7.0 y Microsoft Access 7.0 (y posiblemente en otros productos Microsoft de 32 bits que utilizan Visual Basic para Aplicaciones), puede que los datos de series de las variables Variant recibidos de QMF para Windows no se conviertan de Unicode (utilizado por OLE) a ANSI (utilizado por VBA). Si ocurre esto, sólo se visualiza el primer carácter de la serie. Para solucionar este problema, establezca la variable igual a una serie vacía antes de llamar a la función QMF para Windows que utiliza la variable.

<i>Tipo</i>	Especifica los tipos de objetos de QMF que desea incluir en la lista. Se pueden indicar varios valores para especificar varios tipos de objetos.
-------------	--

Valor	Significado
2048 (RSQ_MASK_QUERIES)	Incluir consultas de QMF en la lista.
1024 (RSQ_MASK_FORMS)	Incluir formatos de QMF en la lista.
512 (RSQ_MASK_PROCS)	Incluir procedimientos de QMF en la lista.
256 (RSQ_MASK_TABLES)	Incluir tablas en la lista.

### Valor devuelto

Cero indica que la función se ha completado correctamente; un valor distinto de cero significa que ha habido un error. Si no se encuentran objetos de QMF coincidentes, el valor devuelto de cero. Si el valor devuelto es distinto de cero, puede llamar a `GetLastErrorString()` o `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` o `GetLastSQLState()` para obtener información adicional del error.

## GetQMFOBJECTListEx()

short GetQMFOBJECTListEx(BSTR *Propietario*, BSTR *Nombre*, short *Índice*)

### Descripción

Esta función devuelve el nombre del objeto de QMF que coincide con los patrones especificados en los parámetros *Propietario* y *Nombre* a los que hace referencia el parámetro *Índice*. Después de llamar a esta función, se puede interrogar a la propiedad *Valor* por el error devuelto.

### Parámetros

Nombre	Descripción
Propietario	Serie que contiene el propietario de los objetos que desea incluir en la lista devuelta.
Nombre	Serie que contiene el nombre de los objetos que desea incluir en la lista devuelta.
Índice	El índice de la lista de objetos de QMF que coinciden con el patrón.

<i>Tipo</i>	Especifica los tipos de objetos de QMF que desea incluir en la lista. Se pueden indicar varios valores para especificar varios tipos de objetos.
-------------	--

Valor	Significado
-------	-------------

2048 (RSQ_MASK_QUERIES)	Incluir consultas de QMF en la lista.
1024 (RSQ_MASK_FORMS)	Incluir formatos de QMF en la lista.
512 (RSQ_MASK_PROCS)	Incluir procedimientos de QMF en la lista.
256 (RSQ_MASK_TABLES)	Incluir tablas en la lista.

### Valor devuelto

Cero indica que la función se ha completado correctamente; un valor distinto de cero significa que ha habido un error. Si no se encuentran objetos de QMF coincidentes, el valor devuelto es RS\_ERROR\_OUTOFRANGE. Si el valor devuelto es distinto de cero, puede llamar a GetLastErrorString() o GetLastErrorType(), GetLastErrorSQLCode(), GetLastErrorSQLError() o GetLastErrorSQLState() para obtener información adicional del error.

## GetQMFProcText()

BSTR GetQMFProcText(BSTR *PropietarioYNombre*)

### Descripción

Esta función devuelve el texto que se ejecuta para el procedimiento especificado, después de la sustitución de variables. Debe utilizar SetProcVariable() para establecer el valor de las variables utilizadas en el procedimiento antes de llamar a esta función.

### Parámetros

Nombre	Descripción
PropietarioYNombre	Serie que contiene el propietario y el nombre, separados por un punto, del objeto que se desea suprimir. Por ejemplo, Juan.Proc2

### Valor devuelto

Serie que contiene el texto del procedimiento que se ha recuperado o NULL si no se ha podido recuperar el procedimiento. Si el valor devuelto es NULL, puede llamar a GetLastErrorString(), GetLastErrorType(), GetLastErrorSQLCode(), GetLastErrorSQLError, o GetLastErrorSQLState() para obtener información adicional del error.

## GetQMFQueryText()

BSTR GetQMFQueryText(BSTR *PropietarioYNombre*)

### Descripción

Esta función recupera el texto de SQL almacenado en la consulta especificada.

### Parámetros

Nombre	Descripción
PropietarioYNombre	Serie que contiene el propietario y el nombre, separados por un punto, del objeto que se desea suprimir. Por ejemplo, Juan.Consulta2

### Valor devuelto

Serie que contiene el texto para la consulta que se ha recuperado o NULL si no se ha podido recuperar la consulta. Si el valor devuelto es NULL, puede llamar a

GetLastErrorString(), GetLastErrorType(), GetLastErrorCode(), GetLastError, o GetLastErrorState() para obtener adicional del error.

## GetQueryText()

BSTR GetQueryText(long *IDConsulta*)

### Descripción

Esta función devuelve el texto de SQL que se ejecuta para la consulta especificada, después de la sustitución de variables. Debe utilizar SetVariable() para establecer el valor de las variables utilizadas en la consulta antes de llamar a esta función.

### Parámetros

Nombre	Descripción
IDConsulta	ID de la consulta devuelto desde InitializeQuery().

**Nota:** El texto de la consulta no está disponible para sentencias de SQL estático. Para los ID de consulta devueltos desde InitializeStaticQuery(), GetQueryText() devuelve un serie vacía.

### Valor devuelto

Si finaliza correctamente, se devuelve una serie que contiene el texto de SQL. Si no finaliza correctamente, se devuelve NULL. Si el valor devuelto es NULL, puede llamar a GetLastErrorString() o GetLastErrorType() para obtener información adicional del error.

## GetQueryVerb()

BSTR GetQueryVerb(long *IDConsulta*)

### Descripción

Esta función devuelve una serie que contiene el verbo de SQL utilizado en la consulta.

### Parámetros

Nombre	Descripción
IDConsulta	ID de la consulta devuelto desde InitializeQuery().

**Nota:** El verbo de la consulta no está disponible para sentencias de SQL estático. Para los ID de consulta devueltos desde InitializeStaticQuery(), GetQueryVerb() devuelve una serie vacía.

### Valor devuelto

Si finaliza correctamente, se devuelve una serie que contiene un verbo de SQL. Si la consulta no finaliza correctamente, se devuelve NULL. Si el valor devuelto es NULL, puede llamar a GetLastErrorString() o GetLastErrorType() para obtener información adicional del error.

## GetResourceLimit()

short GetResourceLimit(short *Recurso*, long\* *Valor*)

## Descripción

Esta función obtiene el límite de recursos solicitados. Deberá llamar a InitializeServer() antes de llamar a esta función, dado que los límites de recursos se manejan según cada servidor.

## Parámetros

Nombre	Descripción
Recurso	Los valores de recurso incluyen:

Valor	Significado
0 (RSR_IDLE_CONNECTION_TIMEOUT)	Tiempo de espera de conexión desocupada (en segundos).
1 (RSR_IDLE_QUERY_TIMEOUT_CANCEL)	Tiempo de espera de consulta desocupada (en segundos).
2 (RSR_IDLE_QUERY_TIMEOUT_WARNING)	Tiempo de espera de consulta desocupada (en segundos). Este es el límite de aviso; no se impone para la API de QMF para Windows.
3 (RSR_SERVER_RESPONSE_TIMEOUT_CANCEL)	Tiempo de espera de servidor (en segundos).
4 (RSR_SERVER_RESPONSE_TIMEOUT_WARNING)	Tiempo de espera de servidor (en segundos). Este es el límite de aviso; no se impone para la API de QMF para Windows.
5 (RSR_MAX_ROWS_TO_FETCH_CANCEL)	Número máximo de filas que se recuperarán.
6 (RSR_MAX_ROWS_TO_FETCH_WARNING)	Número máximo de filas que se recuperarán. Este es el límite de aviso; no se impone para la API de QMF para Windows.
7 (RSR_MAX_BYTES_TO_FETCH_CANCEL)	Número máximo de bytes que se recuperarán.
8 (RSR_MAX_BYTES_TO_FETCH_WARNING)	Número máximo de bytes que se recuperarán. Este es el límite de aviso; no se impone para la API de QMF para Windows.
9 (RSR_MAX_CONNECTIONS)	Número máximo de conexiones permitidas para el servidor de bases de datos.
10 (RSR_ALLOW_SERVER_ACCESS_UI)	¿El acceso al servidor de bases de datos está permitido desde la interfaz de QMF para Windows?

11 (RSR_ALLOW_SERVER_ACCESS_API)	¿EL acceso al servidor de bases de datos está permitido desde la API de QMF para Windows?
12 (RSR_FETCH_ALL_ROWS)	¿Recuperar automáticamente todas las filas?
13 (RSR_CONFIRM_UPDATES)	¿Confirmar actualizaciones del servidor de bases de datos? Esta opción no afecta a la API de QMF para Windows; las actualizaciones de la base de datos no se confirman para la API de QMF para Windows.
14 (RSR_SUMMARY_TRACKING)	¿Está habilitado el seguimiento de objetos resumido?
15 (RSR_DETAILED_TRACKING)	¿Está habilitado el seguimiento de objetos detallado?
16 (RSR_SQL_TRACKING)	¿Está habilitado el seguimiento de texto de SQL?
17 (RSR_ADHOC_TRACKING)	¿Está habilitado el seguimiento de objetos ad hoc?
18 (RSR_ALLOW_ACQUIRE)	¿Está permitido el verbo ACQUIRE de SQL?
19 (RSR_ALLOW_ALTER)	¿Está permitido el verbo ALTER de SQL?
20 (RSR_ALLOW_COMMENT)	¿Está permitido el verbo COMMENT de SQL?
21 (RSR_ALLOW_CREATE)	¿Está permitido el verbo CREATE de SQL?
22 (RSR_ALLOW_DELETE)	¿Está permitido el verbo DELETE de SQL?
23 (RSR_ALLOW_DROP)	¿Está permitido el verbo DROP de SQL?
24 (RSR_ALLOW_EXPLAIN)	¿Está permitido el verbo EXPLAIN de SQL?
25 (RSR_ALLOW_GRANT)	¿Está permitido el verbo GRANT de SQL?
26 (RSR_ALLOW_INSERT)	¿Está permitido el verbo INSERT de SQL?
27 (RSR_ALLOW_LABEL)	¿Está permitido el verbo LABEL de SQL?
28 (RSR_ALLOW_LOCK)	¿Está permitido el verbo LOCK de SQL?
29 (RSR_ALLOW_REVOKE)	¿Está permitido el verbo REVOKE de SQL?
30 (RSR_ALLOW_SELECT)	¿Está permitido el verbo SELECT de SQL?

31 (RSR_ALLOW_SET)	¿Está permitido el verbo SET de SQL?
32 (RSR_ALLOW_SIGNAL)	¿Está permitido el verbo SIGNAL de SQL?
33 (RSR_ALLOW_UPDATE)	¿Está permitido el verbo UPDATE de SQL?
34 (RSR_ALLOW_CALL)	¿Está permitido el verbo CALL de SQL?
35 (RSR_ALLOW_SAVE_DATA)	¿Está permitido el mandato Guardar datos?
36 (RSR_SAVE_DATA_TABLE_SPACE_NAME)	El nombre de espacio de tabla para el mandato Guardar datos
37 (RSR_SAVE_DATA_TABLE_SPACE_NAME_OVERRIDE)	¿Puede el usuario alterar el nombre de espacio de tabla para el mandato Guardar datos?
38 (RSR_ALLOW_BIND_PACKAGE)	¿Permitir vinculación de paquetes?
39 (RSR_DEF_COLLECTION)	Nombre de colección por omisión para vincular paquetes.
40 (RSR_DEF_COLLECTION_OVERRIDE)	¿Puede el usuario alterar el nombre de colección por omisión para vincular paquetes?
41 (RSR_DEF_ISOLATION_LEVEL)	Nivel de aislamiento por omisión para vincular paquetes.
42 (RSR_DEF_ISOLATION_LEVEL_OVERRIDE)	¿Puede el usuario alterar el nivel de aislamiento por omisión para vincular paquetes?
43 (RSR_ALLOW_TABLE_EDIT)	¿Se permite la utilización del editor de tablas?
44 (RSR_ALLOW_EXPORT)	¿Se permite la exportación de datos?
45 (RSR_ALLOW_SAVED_QUERIES_ONLY)	¿Se permite al usuario ejecutar solamente consultas guardadas?
46 (RSR_ALLOW_DROP_PACKAGE)	¿Permitir eliminación de paquetes?
47 (RSR_QUERY_ISOLATION_LEVEL)	Nivel de aislamiento que se debe utilizar al ejecutar las consultas.
48 (RSR_ACCOUNT_STRING)	Serie que especifica información de contabilidad que se deberá pasar al conectar con el servidor de bases de datos.

49 (RSR_ACCOUNT_OVERRIDE)	¿Puede el usuario alterar la serie que especifica información de contabilidad que se debe pasar al conectar con el servidor de bases de datos?
Valor	Puntero a un de tipo Long donde se almacenará el resultado. El resultado es el valor del límite de recursos solicitado. Para valores booleanos, el resultado distinto de cero indica true y cero indica false. Para RSR_SAVE_DATA_TABLE_SPACE_NAME, RSR_DEF_COLLECTION y RSR_ACCOUNT_STRING, se devuelve -1 y se puede interrogar a la propiedad <i>LímiteRecurso</i> acerca del valor de serie devuelto.

### Valor devuelto

Cero indica que la función se ha completado correctamente; un valor distinto de cero significa que ha habido un error. Si el valor devuelto es distinto de cero, puede llamar a `GetLastErrorString()` o `GetLastErrorType()` para obtener información adicional del error.

## GetResourceLimitEx()

short GetResourceLimitEx(short *Recurso*)

### Descripción

Esta función obtiene el límite de recursos solicitados. Deberá llamar a `InitializeServer()` antes de llamar a esta función, dado que los límites de recursos se manejan según cada servidor. Después de llamar a esta función, consulte la propiedad *LímiteRecurso* para conocer el resultado.

### Parámetros

Nombre	Descripción
Recurso	Los valores de recurso son los mismos que los de la llamada a <code>GetResourceLimit()</code> .

**Nota:** El verbo de la consulta no está disponible para sentencias de SQL estático. Para los ID de consulta devueltos desde `InitializeStaticQuery()`, `GetQueryVerb()` devuelve una serie vacía.

### Valor devuelto

Cero indica que la función se ha completado correctamente; un valor distinto de cero significa que ha habido un error. Si el valor devuelto es distinto de cero, puede llamar a `GetLastErrorString()` o `GetLastErrorType()` para obtener información adicional del error.

## GetRowCount()

long GetRowCount(long *IDConsulta*)

### Descripción

Esta función devuelve el número de filas que hay actualmente en el almacenamiento intermedio interno de QMF para Windows. Éste puede ser mayor

que el número de filas recuperadas con `FetchNextRow()` o `FetchNextRows()`, dado que QMF para Windows coloca en el almacenamiento intermedio los datos recibidos de la base de datos.

Esta función devuelve el número de filas ya recuperadas de la base de datos. Si desea recuperar el número total de filas en el conjunto de resultados, puede:

- Llamar a `CompleteQuery()` y obtener todas las filas utilizando `FetchNextRow()` o `FetchNextRows()`.
- Especificar `RecuperaTodas = TRUE` al llamar a `Open()`.

## Parámetros

Nombre	Descripción
IDConsulta	ID de la consulta devuelto desde <code>InitializeQuery()</code> .

## Valor devuelto

Número de filas, si la función es satisfactoria (0 si no se ha recuperado ninguna fila), o -1 si no es satisfactoria. Si obtiene -1, puede llamar a `GetLastErrorString()` o `GetLastErrorType()` para obtener información adicional del error.

## GetServerList()

`short GetServerList(VARIANT* Lista)`

## Descripción

Esta función devuelve una matriz que contiene los nombres de los servidores de bases de datos definidos en el Archivo de definición del servidor (SDF) de QMF para Windows. Debe definir un servidor de bases de datos en el archivo SDF si desea acceder a dicho servidor utilizando la API de QMF para Windows.

## Parámetros

Nombre	Descripción
Lista	Puntero de una variable VARIANT donde se almacenará el resultado. El resultado es una matriz de series (tipo de variante <code>VT_ARRAY</code>   <code>VT_BSTR</code> ), cada una de las cuales contiene el nombre de un servidor de bases de datos. Si no ha definido ningún servidor de bases de datos, el resultado estará vacío (tipo de variante <code>VT_EMPTY</code> ). Deberá inicializar la variable VARIANT correctamente antes de llamar a esta función. Visual Basic efectúa esta acción automáticamente. Los programadores de Visual C++ deben llamar a <code>VariantInit()</code> .

**Nota:** Debido a un problema en Microsoft Excel 7.0 y Microsoft Access 7.0 (y posiblemente en otros productos Microsoft de 32 bits que utilizan Visual Basic para Aplicaciones), puede que los datos de series de las variables Variant recibidos de QMF para Windows no se conviertan de Unicode (utilizado por OLE) a ANSI (utilizado por VBA). Si ocurre esto, sólo se visualiza el primer carácter de la serie. Para solucionar este problema, establezca la variable igual a una serie vacía antes de llamar a la función QMF para Windows que utiliza la variable.

## Valor devuelto

Cero indica que la función se ha completado correctamente; un valor distinto de cero significa que ha habido un error. Si no ha definido ningún servidor de bases

de datos, el valor devuelto es cero. Si el valor devuelto es distinto de cero, puede llamar a `GetLastErrorString()` o `GetLastErrorType()` para obtener información adicional del error.

## GetServerListEx()

short `GetServerListEx(short Índice)`

### Descripción

Esta función recupera el nombre del servidor al que hace referencia el parámetro *Índice*. Después de llamar a esta función, se puede interrogar a la propiedad *Valor* por el valor devuelto.

### Parámetros

Nombre	Descripción
Índice	Índice en la lista de servidores.

### Valor devuelto

Cero indica que la función se ha completado correctamente; `RS_OUTOFRANGE` cuando el índice es mayor que el número de servidores disponibles, un valor distinto de cero significa que ha habido un error. Si no ha definido ningún servidor de bases de datos, el valor devuelto será `RS_OUTOFRANGE`. Si el valor devuelto es distinto de cero, puede llamar a `GetLastErrorString()` o `GetLastErrorType()` para obtener información adicional del error.

## GetStoredProcedureResultSets()

short `GetStoredProcedureResultSets(long IDConsulta, VARIANT* ConjResult)`

### Descripción

Esta función recupera los ID de consulta para los conjuntos de resultados devueltos por el procedimiento almacenado ejecutado con el *IDConsulta* original. Cada ID de consulta devuelto se puede utilizar con `FetchNextRow()` o `FetchNextRows()` para recuperar las filas del conjunto de resultados y `Close()` cuando se alcanza el final de cada conjunto de resultados.

### Parámetros

Nombre	Descripción
IDConsulta	ID de la consulta original devuelto desde <code>InitializeQuery()</code> .
ConjResult	Puntero de una variable <code>VARIANT</code> donde se almacenan los ID de consulta para los conjuntos de resultados. El resultado es una matriz de enteros largos (tipo de variante <code>VT_ARRAY   VT_I4</code> ), en que cada entero es el ID de consulta para los conjuntos de resultados correspondientes. Si el procedimiento almacenado no ha devuelto ningún conjunto de resultados, el resultado estará vacío (tipo de variante <code>VT_EMPTY</code> ). Deberá inicializar la variable <code>VARIANT</code> correctamente antes de llamar a esta función. Visual Basic efectúa esta acción automáticamente. Los programadores de Visual C++ deben llamar a <code>VariantInit()</code> .

### Valor devuelto

Cero indica que la función se ha completado correctamente; un valor distinto de cero significa que ha habido un error. Si el valor devuelto es distinto de cero,

puede llamar a `GetLastErrorString()` o `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` o `GetLastSQLState()` para obtener información adicional del error.

## GetVariables()

short `GetVariables(long IDConsulta, VARIANT* Variables)`

### Descripción

Esta función devuelve una matriz de los nombres de todas las variables en el texto de SQL de la consulta. Deberá asignar valores a estas variables llamando a `SetVariable()` antes de ejecutar la consulta utilizando `Open()` o `Execute()`.

### Parámetros

Nombre	Descripción
IDConsulta	ID de la consulta devuelto desde <code>InitializeQuery()</code> .
Variables	Puntero de una variable <code>VARIANT</code> donde se almacenará el resultado. El resultado es una matriz de series (tipo de variante <code>VT_ARRAY   VT_BSTR</code> ), cada una de las cuales contiene el nombre de una variable. Si la sentencia SQL no contiene variables, el resultado estará vacío (tipo de variante <code>VT_EMPTY</code> ). Deberá inicializar la variable <code>VARIANT</code> correctamente antes de llamar a esta función. Visual Basic efectúa esta acción automáticamente. Los programadores de Visual C++ deben llamar a <code>VariantInit()</code> .

**Nota:** Debido a un problema en Microsoft Excel 7.0 y Microsoft Access 7.0 (y posiblemente en otros productos Microsoft de 32 bits que utilizan Visual Basic para Aplicaciones), puede que los datos de series de las variables Variant recibidos de QMF para Windows no se conviertan de Unicode (utilizado por OLE) a ANSI (utilizado por VBA). Si ocurre esto, sólo se visualiza el primer carácter de la serie. Para solucionar este problema, establezca la variable igual a una serie vacía antes de llamar a la función QMF para Windows que utiliza la variable.

### Valor devuelto

Cero indica que la función se ha completado correctamente; un valor distinto de cero significa que ha habido un error. Si la sentencia SQL no contiene variables, el valor devuelto será `RS_ERROR_NO_DATA` (-1). Si el valor devuelto es distinto de cero, puede llamar a `GetLastErrorString()` o `GetLastErrorType()` para obtener información adicional del error.

## GetVariablesEx()

short `GetVariablesEx(long IDConsulta, short Índice)`

### Descripción

Esta función devuelve el nombre de la variable del texto de SQL de la consulta a la que hace referencia el parámetro *Índice*. Después de llamar a esta función, se puede interrogar a la propiedad *Valor* por el valor devuelto. Deberá asignar valores a esta variable (y a todas las demás del texto de SQL) llamando a `SetVariable()` antes de ejecutar la consulta utilizando `Open()` o `Execute()`.

### Parámetros

Nombre	Descripción
IDConsulta	ID de la consulta devuelto desde <code>InitializeQuery()</code> .

Índice	Índice en la lista interna de variables. Consulte la propiedad <i>Valor</i> para obtener la serie que se corresponde con el índice pasado. Si la sentencia SQL no contiene variables, la función devolverá RS_ERROR_NO_DATA.
--------	--

### Valor devuelto

Cero indica que la función se ha completado correctamente; un valor distinto de cero significa que ha habido un error. Si la sentencia SQL no contiene variables, el valor devuelto será RS\_ERROR\_NO\_DATA (-1). Si el valor devuelto es distinto de cero, puede llamar a `GetLastErrorString()` o `GetLastErrorType()` para obtener información adicional del error.

## InitializeProc()

`long InitializeProc(short TipoOrigen, BSTR Origen)`

### Descripción

Esta función establece el texto que se desea utilizar en el procedimiento. Se puede pasar el texto como parámetro en esta función, leerlo desde un archivo de texto u obtenerlo de un procedimiento existente.

### Parámetros

Nombre	Descripción
TipoOrigen	Especifica el origen del texto del procedimiento.

Valor	Significado
0 (RSS_STRING)	El texto figura en el parámetro <i>Origen</i> .
2 (RSS_FILE)	El texto figura en el archivo de texto cuyo nombre está especificado en el parámetro <i>Origen</i> .
3 (RSS_QMFPROC)	El texto figura en el procedimiento cuyo propietario y nombre están especificados en el parámetro <i>Origen</i> .

<i>Origen</i>	Serie que contiene el texto, o bien el propietario y el nombre (Propietario.Nombre) del procedimiento o bien el nombre de un archivo que contiene el procedimiento.
---------------	---

### Valor devuelto

Si la función es satisfactoria, el ID de procedimiento (IDProc). Si no es satisfactoria, -1. Deberá utilizar este valor en todas llamadas de interfaz que necesiten el parámetro *IDProc*.

## InitializeQuery()

`long InitializeQuery(short TipoOrigen, BSTR Origen)`

### Descripción

Esta función establece el texto que desea utilizar en la consulta. Se puede pasar el texto de SQL como parámetro en esta función, leerlo desde un archivo de texto u obtenerlo de un procedimiento existente. Llame a `Close()` cuando haya terminado con esta consulta.

## Parámetros

Nombre	Descripción
TipoOrigen	Especifica el origen del texto de la sentencia SQL.

Valor	Significado
0 (RSS_STRING)	El texto de SQL figura en el parámetro <i>Origen</i> .
1 (RSS_QMFQUERY)	El texto de SQL figura en la consulta cuyo propietario y nombre están especificados en el parámetro <i>Origen</i> .
2 (RSS_FILE)	El texto de SQL figura en el archivo de texto cuyo nombre se especifica en el parámetro <i>Origen</i> .

### Valor devuelto

Si la función es satisfactoria, el ID de la consulta. Si no es satisfactoria, -1. Deberá utilizar este valor en todas las llamadas de interfaz que necesiten el parámetro *Consulta*.

## InitializeServer()

short InitializeServer(BSTR *NombServ*, BSTR *IDUsuario*, BSTR *Contraseña*, BOOL *DialogOblig*, [VARIANT *Cuenta*], [VARIANT *SuprimDiálogo*])

### Descripción

Esta función inicializa una conexión a un servidor de bases de datos. Deberá llamar a esta función antes de llamar a cualquier otra función de la API de QMF para Windows. Se puede llamar a esta función varias veces. Sin embargo, si llama a esta función y no se finaliza efectuando una llamada a Commit() o Rollback(), se obtendrá una retrotracción implícita.

## Parámetros

Nombre	Descripción
NombServ	Serie que contiene el nombre del servidor de bases de datos que desea utilizar. Este nombre debe coincidir con alguno de los nombres definidos en el archivo de definición del servidor de QMF para Windows. Llame a GetServerList() para recuperar una lista de servidores válidos.
IDUsuario	Serie que contiene el ID de usuario que desea utilizar. Si IDUsuario es NULL o una serie vacía, QMF para Windows intentará utilizar el IDUsuario de la consulta más reciente, si está disponible. De lo contrario, QMF para Windows mostrará el recuadro de diálogo Información del usuario para obtener un ID de usuario y una contraseña.
Contraseña	Serie que contiene la contraseña para el usuario especificado. Si Contraseña es NULL o una serie vacía, QMF para Windows intentará utilizar una contraseña memorizada si está disponible (necesita Windows para Trabajo en Grupo). Si no hay ninguna contraseña disponible, QMF para Windows mostrará el recuadro de diálogo Información del usuario para obtener una contraseña.

DialogOblig	Un valor distinto de cero indica que QMF para Windows mostrará el recuadro de diálogo Información del usuario independientemente de que se hayan especificado un ID de usuario y una contraseña. Esto permite al usuario cambiar la información antes de que se utilice. Cero indica que QMF para Windows debe mostrar el recuadro de diálogo Información del usuario solamente cuando sea necesario.
Cuenta	Opcionalmente, una serie que especifica información de contabilidad que se pasará al servidor al conectar con el mismo. El servidor puede utilizar esta información en un sistema de contabilidad de trabajos.
SuprimDiálogo	Un valor distinto de cero indica que QMF para Windows no visualizará el recuadro de diálogo Información del usuario, aun si no se ha especificado un ID de usuario y contraseña. Esta opción es útil cuando se ejecuta en un entorno donde no está presente un usuario para responder al recuadro de diálogo Información del usuario, por ejemplo, en un servidor web.

### Valor devuelto

Cero indica que la función se ha completado correctamente; un valor distinto de cero significa que ha habido un error. Si el valor devuelto es distinto de cero, puede llamar a `GetLastErrorString()` o `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` o `GetLastSQLState()` para obtener información adicional del error.

### Temas relacionados

`SetParent()`

## InitializeStaticQuery()

`long InitializeStaticQuery(BSTR NombreColección, BSTR NombrePaquete, BSTR SímboloConsist, short NúmSección)`

### Descripción

Esta función especifica la sección de un paquete que se desea ejecutar como consulta estática.

### Parámetros

Nombre	Descripción
NombreColección	Nombre de una colección vinculada anteriormente.
NombrePaquete	Nombre de un paquete vinculado anteriormente.
SímboloConsist	Símbolo utilizado por la colección y el paquete nombrados más arriba.
NúmSección	Número de sección de la sentencia contenida en la colección y el paquete que desea utilizar.

### Valor devuelto

Si la función es satisfactoria, el ID de la consulta. Si no es satisfactoria, -1. Deberá utilizar este valor en todas las llamadas de interfaz que necesiten el parámetro *IDConsulta*.

## IsStatic()

`BOOL IsStatic(long IDConsulta)`

## Descripción

Esta función determina si el ID de consulta especificado se refiere a una consulta estática o a una consulta dinámica.

## Parámetros

Nombre	Descripción
IDConsulta	ID de la consulta devuelta desde InitializeQuery() o InitializeStaticQuery().

## Valor devuelto

Devuelve un valor distinto de cero si es satisfactoria y *IDConsulta* hace referencia a una consulta estática, de lo contrario es cero.

## Open()

short Open(long *IDConsulta*, long *LímFilas*, BOOL *RecuperaTodas*)

## Descripción

Utilice esta función para ejecutar una consulta que utiliza el verbo SELECT, mediante la apertura de un cursor en la base de datos para la consulta. Utilice FetchNextRow() o FetchNextRows() para recuperar los datos para la consulta y llame a Close() cuando haya terminado. Si QMF para Windows está configurado para recuperar automáticamente todas las filas (vea RSR\_FETCHALLROWS en la descripción de GetResourceLimit()) o el parámetro RecuperaTodas tiene un valor distinto de cero, QMF para Windows recuperará todas las filas del conjunto de resultados en su almacenamiento intermedio interno antes de volver de esta llamada.

**Nota:** El nombre de esta función está en conflicto con la palabra clave Open de Microsoft Access 2.0. Si está utilizando MS Access 2.0, ponga el nombre de función entre corchetes [ ].

**Nota:** Utilice esta función solamente en las sentencias que contengan el verbo SELECT de SQL. Para sentencias que contengan cualquier otro verbo, por ejemplo, SET, llame a Execute() en su lugar. Para determinar el verbo utilizado por una consulta, llame a GetQueryVerb().

## Parámetros

Nombre	Descripción
<i>IDConsulta</i>	ID de la consulta devuelto desde InitializeQuery().
<i>LímFilas</i>	Número que indica el número máximo de filas a recuperar de la base de datos. Cero indica que no se impone ningún límite excepto para el límite de filas establecido por el programa del Administrador de QMF para Windows.
<i>RecuperaTodas</i>	Valor booleano que indica si todas las filas del conjunto de resultados se recuperan automáticamente o no del almacenamiento intermedio de QMF para Windows. Si el valor es distinto de cero, se recuperan automáticamente todas las filas, cerrando el cursor y liberando la base de datos para que la utilicen otros usuarios. Es igual a la llamada a CompleteQuery().

## Valor devuelto

Cero indica que la función se ha completado correctamente; un valor distinto de cero significa que ha habido un error. Si el valor devuelto es distinto de cero,

puede llamar a `GetLastErrorString()` o `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` o `GetLastSQLState()` para obtener información adicional del error.

## Prepare()

short Prepare(long *IDConsulta*)

### Descripción

Esta función prepara la consulta especificada por *IDConsulta*. La sentencia la examina el servidor de bases de datos, comprobando la existencia del objeto, las autorizaciones necesarias, etc. Si la consulta es una sentencia SELECT, la información acerca de las columnas devueltas por la sentencia está disponible después de completar `Prepare()`.

### Parámetros

Nombre	Descripción
<i>IDConsulta</i>	ID de la consulta devuelto desde <code>InitializeQuery()</code> .

### Valor devuelto

Cero indica que la función se ha completado correctamente; un valor distinto de cero significa que ha habido un error. Si el valor devuelto es distinto de cero, puede llamar a `GetLastErrorString()` o `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` o `GetLastSQLState()` para obtener información adicional del error.

### Temas relacionados

`Execute()`  
`Open()`

## PrintReport()

short PrintReport(long *IDConsulta*, short *TipoOrigen*, BSTR *Origen*, BSTR *NombreArchSalida*, short *LongPágina*, short *AnchoPág*, BOOL *IncluirFechaHora*, BOOL *IncluirNúmPág*, [VARIANT *Formato*], [VARIANT *UsarDefinPágForm*])

### Descripción

`PrintReport()` es sinónimo de la función `ExportReport()`.

## ReinitializeServer()

short ReinitializeServer()

### Descripción

Esta función reinicializa la conexión con el servidor de bases de datos. Normalmente solamente es necesario llamar a esta función si una de las otras funciones de la API de QMF para Windows devuelve un error. La llamada a esta función origina una retrotracción implícita, la cual cierra todos los cursores abiertos e invalida todos los ID de consulta pendientes.

### Valor devuelto

Cero indica que la función se ha completado correctamente; un valor distinto de cero significa que ha habido un error. Si el valor devuelto es distinto de cero, puede llamar a `GetLastErrorString()` o `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` o `GetLastSQLState()` para obtener información adicional del error.

## Rollback()

short Rollback()

### Descripción

Esta función cancela todos los cambios efectuados en la unidad de trabajo actual, finaliza la unidad de trabajo actual, cierra los cursores abiertos e invalida todos los ID de consulta pendientes.

**Nota:** El nombre de esta función está en conflicto con la palabra clave Rollback de Microsoft Access 2.0. Si está utilizando MS Access 2.0, ponga el nombre de función entre corchetes [ ].

**Nota:** La retrotracción solamente afecta a los cambios de SQL ejecutados mediante llamadas a Open() o Execute(). La retrotracción no afecta a los cambios efectuados por otras funciones de la API de QMF para Windows, por ejemplo, FastSaveData(), SaveData() o DeleteQMFObject().

### Valor devuelto

Cero indica que la función se ha completado correctamente; un valor distinto de cero significa que ha habido un error. Si el valor devuelto es distinto de cero, puede llamar a GetLastErrorString() o GetLastErrorType(), GetLastErrorCode(), GetLastErrorError() o GetLastErrorState() para obtener información adicional del error.

### Temas relacionados

Commit()

## RunProc()

short RunProc(long IDProc)

### Descripción

Esta función ejecuta el procedimiento especificado. El procedimiento se ejecuta hasta su finalización o hasta que se produce un error. No se puede acceder a ningún resultado del procedimiento (por ejemplo, datos de una consulta que se ha ejecutado) mediante esta interfaz de programación. Sin embargo, los archivos exportados o datos guardados por el procedimiento están disponibles después de la ejecución.

### Parámetros

Nombre	Descripción
IDProc	ID de procedimiento devuelto desde InitializeProc().

### Valor devuelto

Cero indica que la función se ha completado correctamente; un valor distinto de cero significa que ha habido un error. Si el valor devuelto es distinto de cero, puede llamar a GetLastErrorString() o GetLastErrorType(), GetLastErrorCode(), GetLastErrorError() o GetLastErrorState() para obtener información adicional del error.

## SaveData()

short SaveData(long IDConsulta, long PrimeraFila, long PrimeraCol, long ÚltimaFila, long ÚltimaCol, BOOL Sustituir, BSTR NombreTabla, BSTR NombreEspacioTabla, BSTR NombServ, BSTR IDUsuario, BSTR Contraseña, BOOL DialogOblig, [VARIANT Cuenta], [VARIANT Comentario], [VARIANT ÁmbitoConfirm])

## Descripción

Esta función guarda el rango especificado de filas y columnas en el espacio de tabla especificada. Deberá llamar a CompleteQuery() antes de llamar a esta función si no se han recuperado datos de fila para todas las filas que desea guardar en la tabla. Si intenta guardar filas que no se han recuperado de la base de datos, fallará la operación de guardar. Si ya existe la tabla, los datos nuevos deben tener el mismo número y los mismos tipos de columnas que la tabla existente.

Esta función opera en una unidad de trabajo independiente de las de otras funciones de la API y sus resultados se confirman automáticamente. La llamada a Commit() o Rollback() no afectará a los cambios efectuados utilizando esta función.

## Parámetros

Nombre	Descripción
<i>IDConsulta</i>	ID de la consulta devuelto desde InitializeQuery().
PrimeraFila	Primera fila que desea incluir en la operación de guardar. El valor de la primera fila de un conjunto de resultados es 0.
PrimeraCol	La primera columna que desea incluir en la operación de guardar. El valor de la primera columna en un conjunto de resultados es cero.
ÚltimaFila	La última fila que desea incluir en la operación de guardar o -1 si se incluyen todas las filas. El valor de la última fila de un conjunto de resultados es el número total de filas menos uno.
ÚltimaCol	La última columna que desea incluir en la operación de guardar o -1 si se incluyen todas las columnas. El valor de la última columna de un conjunto de resultados es el número total de columnas menos uno.
Sustituir	Un valor distinto de cero indica que los datos especificados sustituirán los datos existentes en la tabla. Cero indica que los datos especificados se añadirán a los datos existentes en la tabla.
NombreTabla	Nombre de la tabla donde se almacenarán los datos. Si la tabla no existe, se creará.
NombreEspacioTabla	Nombre del espacio de tabla en el que existe o se creará la tabla. Si <i>NombreEspacioTabla</i> es NULL o una serie vacía, se utilizará el espacio de tabla por omisión. Si ha configurado QMF para Windows para utilizar siempre el espacio de tabla por omisión (vea RSR_SDDIFFERENTTS en la descripción de GetResourceLimit()), este parámetro se ignorará.
NombServ	Nombre del servidor de bases de datos donde se almacenará la tabla. Si <i>NombServ</i> es NULL o una serie vacía, se utilizará el servidor que especifique en la llamada a InitializeServer() y se ignorarán <i>IDUsuario</i> , <i>Contraseña</i> , <i>DialogOblig</i> y <i>Cuenta</i> .
<i>IDUsuario</i>	Si ha especificado un servidor diferente en <i>NombServ</i> , <i>IDUsuario</i> es el ID de usuario especificado por ese servidor. Si no se especifica un ID de usuario, QMF para Windows utilizará el que se ha especificado por última vez para este servidor si está disponible o mostrará un recuadro de diálogo si no hay ninguno disponible. Es parámetro se ignora si <i>NombServ</i> es NULL o es una serie vacía.
Contraseña	Si ha especificado un servidor diferente en <i>NombServ</i> , <i>Contraseña</i> es la contraseña utilizada para este servidor. Si no se especifica una contraseña, QMF para Windows utilizará la última contraseña especificada para este servidor si está disponible o visualizará un recuadro de diálogo si no hay ninguna disponible. Es parámetro se ignora si <i>NombServ</i> es NULL o es una serie vacía.

DialogOblig	Si se ha especificado un servidor diferente en <i>NombServ</i> , el valor distinto de cero fuerza a QMF para Windows para que muestre un recuadro de diálogo que solicite información de inicio de sesión, incluso si se ha especificado un ID de usuario y una contraseña o éstos están disponibles de algún otro modo. Cero indica que QMF para Windows mostrará este recuadro de diálogo solamente si es necesario. Es parámetro se ignora si <i>NombServ</i> es NULL o es una serie vacía.
Cuenta	Si se ha especificado un servidor diferente en <i>NombServ</i> , opcionalmente, una serie que especifica información de contabilidad para pasar al servidor al conectar con el mismo. El servidor puede utilizar esta información en un sistema de contabilidad de trabajos. Es parámetro se ignora si <i>NombServ</i> es NULL o es una serie vacía.
Comentario	Opcionalmente, una serie que especifica un comentario para la tabla en la que se han guardado los datos.
ÁmbitoConfirm	Opcionalmente, número de filas que se insertarán en la tabla de una sola vez antes de confirmar la unidad de trabajo. Al especificar cero se indica que todas las filas se deben insertar antes de la confirmación. Si se especifica 10 (por ejemplo), indica que se debe realizar la confirmación cada vez que se inserten diez filas.

### Valor devuelto

Cero indica que la función se ha completado correctamente; un valor distinto de cero significa que ha habido un error. Si el valor devuelto es distinto de cero, puede llamar a `GetLastErrorString()` o `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` o `GetLastSQLState()` para obtener información adicional del error. Si el conjunto de resultados está vacío o no se recupera ninguna fila de la base de datos, se devolverá un valor distinto de cero a menos que `FirstRow=0` y `LastRow=-1`. En este caso, se devolverá cero y se creará una tabla vacía.

## SaveQMFPProc()

short SaveQMFPProc(BSTR *PropietarioYNombre*, BSTR *Texto*, BSTR *Comentario*, BOOL *Sustituir*, BOOL *Compartir*)

### Descripción

Esta función guarda el procedimiento en un servidor de bases de datos.

### Parámetros

Nombre	Descripción
<i>PropietarioYNombre</i>	Serie que contiene el propietario y el nombre, separados por un punto, del procedimiento que desea guardar. Por ejemplo, Juan.Proc2
Texto	Serie que contiene el texto que desea guardar en el procedimiento.
Comentario	Serie que contiene comentarios que desea guardar con el procedimiento. Si no hay comentarios, pase este parámetro como un serie vacía o NULL.
Sustituir	Un valor distinto de cero sustituye el procedimiento existente con el mismo nombre. Cero termina anormalmente la operación si hay un procedimiento existente con el mismo nombre.
Compartir	Un valor distinto de cero comparte el procedimiento con otros usuarios. Cero no comparte el procedimiento con otros usuarios.

### Valor devuelto

Cero indica que la función se ha completado correctamente; un valor distinto de cero significa que ha habido un error. Si el valor devuelto es distinto de cero, puede llamar a `GetLastErrorString()` o `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` o `GetLastSQLState()` para obtener información adicional del error.

## SaveQMFQuery()

short SaveQMFQuery(BSTR *PropietarioYNombre*, BSTR *Texto*, BSTR *Comentario*, BOOL *Sustituir*, BOOL *Compartir*)

### Descripción

Esta función guarda una consulta en el servidor de bases de datos.

### Parámetros

Nombre	Descripción
<i>PropietarioYNombre</i>	Serie que contiene el propietario y el nombre, separados por un punto, de la consulta que desea guardar. Por ejemplo, Juan.Consulta2
<i>Texto</i>	Serie que contiene el texto que desea guardar en la consulta.
<i>Comentario</i>	Serie que contiene comentarios que desea guardar con la consulta. Si no hay comentarios, pase este parámetro como un serie vacía o NULL.
<i>Sustituir</i>	Un valor distinto de cero sustituye una consulta existente con el mismo nombre. Cero termina anormalmente la operación si existe una consulta con el mismo nombre.
<i>Compartir</i>	Un valor distinto de cero comparte la consulta con otros usuarios. Cero no comparte la consulta con otros usuarios.

### Valor devuelto

Cero indica que la función se ha completado correctamente; un valor distinto de cero significa que ha habido un error. Si el valor devuelto es distinto de cero, puede llamar a `GetLastErrorString()` o `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` o `GetLastSQLState()` para obtener información adicional del error.

## SetBindOption()

short SetBindOption(BSTR *NombreColección*, BSTR *NombrePaquete*, short *Opción*, short *Valor*)

### Descripción

Esta función establece opciones para la colección y el paquete antes de llamar a `EndBind()`.

### Parámetros

Nombre	Descripción
<i>NombreColección</i>	ID de colección del paquete para el que desea establecer la opción.
<i>NombrePaquete</i>	Nombre del paquete para el que desea establecer la opción.
<i>Opción</i>	Una de las opciones listadas más abajo.

Valor	Un valor distinto de cero sustituye una consulta existente con el mismo nombre. Cero termina anormalmente la operación si existe una consulta con el mismo nombre.
Compartir	Uno de los valores listados más abajo para la opción especificada.

Los significados y valores para las diversas opciones son los siguientes:

Opción	Significado	Descripción
DDM_PKGRPLOPT(0x211C)	Distintivo que especifica si se debe sustituir o no un paquete existente con el mismo ID de colección y nombre.	DDM_PKGRPLALW (0x241F) Sí DDM_PKGRPLNA (0x2420) No
DDM_STTDECDEL(0x2121)	Delimitador utilizado para la coma decimal en las sentencias de SQL del paquete.	DDM_DECDELPRD (0x243C) Punto DDM_DECDELCMA (0x243D) Coma
DDM_STTSTRDEL(0x2120)	Delimitador utilizado para los valores de serie en las sentencias de SQL del paquete.	DDM_STRDELAP (0x2426) Apóstrofo DDM_STRDELDQ (0x2427) Comillas dobles
DDM_PKGISOLVL(0x2124)	Nivel de aislamiento para el paquete.	DDM_ISOLVLALL (0x2443) Todo DDM_ISOLVLCHG (0x2441) Cambiar DDM_ISOLVLCS (0x2442) Estabilidad del cursor DDM_ISOLVLNC (0x2445) Sin confirmar DDM_ISOLVLR (0x2444) Lectura repetitiva
DDM_PKGATHOPT(0x211E)	Distintivo que especifica si se deben mantener o no las autoridades existentes en el paquete.	DDM_PKGATHKP (0x2425) Mantener DDM_PKGATHRVK (0x2424) Revocar
DDM_QRYBLKCTL(0x2132)	Método a utilizar cuando se buscan filas de datos para consultas en el paquete.	DDM_FIXROWPRC (0x2418) Una fila a la vez DDM_LMTBLKPRC (0x2417) Un bloque a la vez
DDM_RDBRLSOPT(0x2129)	Indica cuándo se deben liberar recursos de la base de datos obtenidos al ejecutar el paquete.	DDM_RDBRLSCMM (0x2438) Confirmar DDM_RDBRLSCNV (0x2439) Desasignación de conversación
DDM_STTDATEFMT(0x2122)	Formato para los valores de fecha recuperados.	DDM_ISODATEFMT (0x2429) ISO DDM_USADATEFMT (0x242A) EE.UU. DDM_EURDATEFMT (0x242B) Europeo DDM_JISDATEFMT (0x242C) Estándar industrial japonés

DDM_STTIMFMT(0x2123)	Formato para los valores de hora recuperados.	DDM_ISOTIMFMT (0x242E) ISO DDM_USATIMFMT (0x242F) EE.UU. DDM_EURTIMFMT (0x2430) Europeo DDM_JISTIMFMT (0x2431) Estándar industrial japonés
----------------------	---	---

### Valor devuelto

Cero indica que la función se ha completado correctamente; un valor distinto de cero significa que ha habido un error. Si el valor devuelto es distinto de cero, puede llamar a `GetLastErrorString()` o `GetLastErrorType()` para obtener información adicional del error.

## SetBindOwner()

short SetBindOwner(BSTR *NombreColección*, BSTR *NombrePaquete*, BSTR *IDPropietario*)

### Descripción

Esta función le permite especificar un propietario diferente del ID de usuario para el paquete que está vinculando. Puede que esta especificación sea necesaria si el ID de usuario no tiene las autorizaciones necesarias para vincular el paquete, pero el propietario especificado sí las tiene.

### Parámetros

Nombre	Descripción
NombreColección	ID de colección del paquete para el que desea especificar el propietario.
NombrePaquete	Nombre del paquete para el que desea especificar el propietario.
Comentario	Serie que contiene comentarios que desea guardar con la consulta. Si no hay comentarios, pase este parámetro como un serie vacía o NULL.
IDPropietario	ID de propietario deseado para el paquete que está vinculando.

### Valor devuelto

Cero indica que la función se ha completado correctamente; un valor distinto de cero significa que ha habido un error. Si el valor devuelto es distinto de cero, puede llamar a `GetLastErrorString()` o `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` o `GetLastSQLState()` para obtener información adicional del error.

## SetBusyWindowButton()

void SetBusyWindowButton(BSTR *Texto*)

### Descripción

Esta función especifica el texto visualizado en el botón Cancelar de la ventana de proceso en curso.

### Parámetros

Nombre	Descripción
--------	-------------

---

Texto	Serie que especifica el texto visualizado en el botón Cancelar de la ventana de proceso en curso. El valor por omisión es "Cancelar". Si especifica una serie vacía, se oculta el pulsador. Independientemente del texto especificado, el pulsador siempre cancela la operación o cierra la ventana.
-------	--

---

### Valor devuelto

Ninguno.

### Temas relacionados

SetBusyWindowMessage()

SetBusyWindowMode()

SetBusyWindowTitle()

ShowBusyWindow()

## SetBusyWindowMessage()

void SetBusyWindowMessage(BSTR *Mensaje*)

### Descripción

Esta función especifica el texto visualizado en el área de mensajes de la ventana de proceso en curso.

### Parámetros

---

Nombre	Descripción
Mensaje	Serie que especifica el texto visualizado en el área de mensajes de la ventana de proceso en curso.

---

### Valor devuelto

Ninguno.

### Temas relacionados

SetBusyWindowButton()

SetBusyWindowMode()

SetBusyWindowTitle()

ShowBusyWindow()

## SetBusyWindowMode()

void SetBusyWindowMode(short *Modo*)

### Descripción

Esta función determina si QMF para Windows muestra o no la ventana de proceso en curso. La ventana de proceso en curso es útil para proporcionar información de retorno al usuario y para permitir al usuario cancelar una acción pendiente de la base de datos. Los cambios entran en vigor la vez siguiente que QMF para Windows efectúa una operación que hace que la ventana de proceso en curso se visualice o se oculte.

### Parámetros

---

Nombre	Descripción
Modo	Especifica cuándo QMF para Windows muestra la ventana de proceso en curso.

---

Valor	Significado
0 (RSM_NEVER)	La ventana no se visualiza. Es el valor por omisión.
1 (RSM_WHENBUSY)	La ventana se visualiza cuando QMF para Windows está ocupado comunicándose con la base de datos. QMF para Windows muestra automáticamente esta ventana cuando se oportuno.
2 (RSM_CLIENTCONTROLLED)	La ventana se visualiza después de llamar a ShowBusyWindow(TRUE) y hasta que llame a ShowBusyWindow(FALSE). El cliente determina cuándo se muestra la ventana.

### Valor devuelto

Cero indica que la función se ha completado correctamente; un valor distinto de cero significa que ha habido un error. Si el valor devuelto es distinto de cero, puede llamar a GetLastErrorString() o GetLastErrorType() para obtener información adicional del error.

### Temas relacionados

SetBusyWindowButton()  
SetBusyWindowMessage()  
SetBusyWindowTitle()  
SetParent()  
ShowBusyWindow()

## SetBusyWindowTitle()

void SetBusyWindowTitle(BSTR *Título*)

### Descripción

Esta función especifica el texto visualizado en la barra de título de la ventana de proceso en curso.

### Parámetros

Nombre	Descripción
Título	Serie que especifica el texto visualizado en la barra de título de la ventana de proceso en curso.

### Valor devuelto

Ninguno.

### Temas relacionados

SetBusyWindowButton()  
SetBusyWindowMode()  
SetBusyWindowMessage()  
ShowBusyWindow()

## SetGlobalVariable()

short SetGlobalVariable(BSTR *Nombre*, BSTR *Valor*)

### Descripción

Esta función asigna un valor a la variable global especificada. Este valor está disponible para su utilización con consultas, formatos y procedimientos.

## Parámetros

Nombre	Descripción
Nombre	Serie que contiene el nombre de la variable que desea establecer.
Valor	Serie que contiene el valor que desea asignar a la variable especificada.

### Valor devuelto

Cero indica que la función se ha completado correctamente; un valor distinto de cero significa que ha habido un error. Si el valor devuelto es distinto de cero, puede llamar a `GetLastErrorString()` o `GetLastErrorType()` para obtener información adicional del error.

## SetHostVariable()

`short SetHostVariable(long IDConsulta, VARIANT Índice, VARIANT Valor)`

### Descripción

Esta función asigna un valor a la variable del sistema principal especificada a la que hace referencia la consulta. La consulta debe ser una consulta estática que haga referencia a las variables del sistema principal (o bien almacenadas con la consulta QMF o creadas mediante `AddHostVariable()`). *Índice* puede especificar el índice numérico de la variable del sistema principal o el nombre de la variable del sistema principal.

## Parámetros

Nombre	Descripción
IDConsulta	ID de la consulta devuelto desde <code>InitializeStaticQuery()</code> .
Índice	Número (tipo de variante <code>VT_I2</code> ) que especifica el índice de la variable del sistema principal en la consulta o una serie (tipo de variante <code>VT_BSTR</code> ) que especifica el nombre de la variable del sistema principal.
Valor	Valor de la variable del sistema principal. Para especificar un valor nulo, el tipo de la variante debería estar establecido en <code>VT_EMPTY</code> .

### Valor devuelto

Cero indica que la función se ha completado correctamente; un valor distinto de cero significa que ha habido un error. Si el valor devuelto es distinto de cero, puede llamar a `GetLastErrorString()` o `GetLastErrorType()` para obtener información adicional del error.

## SetOption()

`short SetOption(short Modo, VARIANT Valor)`

### Descripción

Esta función establece el valor de la opción especificado en QMF para Windows. Para algunas opciones, los cambios no entrarán en vigor hasta que se reinicie QMF para Windows. En condiciones normales, QMF para Windows no se reiniciará hasta que se hayan destruido todas las instancias del objeto de la API de QMF para Windows.

## Parámetros

Nombre	Descripción
--------	-------------

Opción	Especifica qué opción se debe establecer:
Valor	Significado
0 (RSO_SERVER_DEFINITION_FILE)	Nombre de archivo de definición de servidor.
1 (RSO_CPIC_DLL)	Nombre de archivo de la DLL del proveedor de CPI-C.
2 (RSO_CPIC_TIMEOUT_WARNING)	Tiempo de espera de aviso de CPI-C (en segundos). Este límite no se utiliza para la API de QMF para Windows.
3 (RSO_CPIC_TIMEOUT_CANCEL)	Tiempo de espera de cancelación de CPI-C (en segundos).
4 (RSO_TCP_TIMEOUT_WARNING)	Tiempo de espera de cancelación de TCP (en segundos). Este límite no se utiliza para la API de QMF para Windows.
5 (RSO_TCP_TIMEOUT_CANCEL)	Tiempo de espera de cancelación de TCP (en segundos).
6 (RSO_DISPLAY_NULLS_STRING)	Serie utilizada para visualizar valores nulos.
7 (RSO_ENTER_NULLS_STRING)	Serie utilizada para entrar valores nulos.
8 (RSO_ENTER_DEFAULTS_STRING)	Serie utilizada para entrar valores por omisión.
9 (RSO_TRACE_FILE_1)	Rastrear el nombre de archivo 1.
10 (RSO_TRACE_FILE_2)	Rastrear el nombre de archivo 2.
11 (RSO_TCP_TRACE_LEVEL)	Nivel de rastreo de TCP.
12 (RSO_CPIC_TRACE_LEVEL)	Nivel de rastreo de CPI-C.
13 (RSO_DDM_TRACE_LEVEL)	Nivel de rastreo de DDM.
Nombre	Descripción
Valor	Valor en el que se establecerá la opción.

### Valor devuelto

Cero indica que la función se ha completado correctamente; un valor distinto de cero significa que ha habido un error. Si el valor devuelto es distinto de cero, puede llamar a `GetLastErrorMessage()` o `GetLastErrorType()` para obtener información adicional del error.

### Temas relacionados

`GetOption()`

## SetParent()

`short SetParent(long VentanaPadre)`

### Descripción

Esta función establece la ventana padre para los diálogos. Normalmente, cuando QMF para Windows muestra un diálogo (en la ventana de proceso en curso o en el recuadro de diálogo Información del usuario), dicho diálogo se centra y permanece fijo en la ventana principal de QMF para Windows. Esta función le permite forzar que los diálogos de QMF para Windows se centren en la ventana de la aplicación cliente y permanezcan fijos.

## Parámetros

Nombre	Descripción
VentanaPadre	La HWND de la nueva ventana padre. Especifique NULL para utilizar la ventana principal de QMF para Windows como la ventana padre.

### Valor devuelto

Cero indica que la función se ha completado correctamente; un valor distinto de cero significa que ha habido un error. Si el valor devuelto es distinto de cero, puede llamar a `GetLastErrorString()` o `GetLastErrorType()` para obtener información adicional del error.

### Temas relacionados

`ShowBusyWindow()`

## SetProcVariable()

`short SetProcVariable(long IDProc, BSTR Nombre, BSTR Valor)`

### Descripción

Esta función asigna un valor a la variable especificada. Este valor sustituye a la variable antes de ejecutar el procedimiento. Si el procedimiento contiene una o más variables, deberá llamar a esta función para establecer los valores de variable antes de llamar a `RunProc()`.

### Parámetros

Nombre	Descripción
IDProc	ID de procedimiento devuelto desde <code>InitializeProc()</code> .
Nombre	Serie que contiene el nombre de la variable que desea establecer.
Valor	Serie que contiene el valor que desea asignar a la variable especificada.

### Valor devuelto

Cero indica que la función se ha completado correctamente; un valor distinto de cero significa que ha habido un error. Si el valor devuelto es distinto de cero, puede llamar a `GetLastErrorString()` o `GetLastErrorType()` para obtener información adicional del error.

## SetVariable()

`short SetVariable(long IDConsulta, BSTR Nombre, BSTR Valor)`

### Descripción

Esta función asigna un valor a la variable especificada. Este valor sustituye a la variable antes de ejecutar la sentencia SQL. Si la sentencia SQL contiene una o más variables, deberá llamar a esta función para establecer los valores de variable antes de llamar a `Open()` o `Execute()`.

### Parámetros

Nombre	Descripción
IDConsulta	ID de la consulta devuelto desde <code>InitializeQuery()</code> .
Nombre	Serie que contiene el nombre de la variable que desea establecer.

Valor	Serie que contiene el valor que desea asignar a la variable especificada.
-------	---

### Valor devuelto

Cero indica que la función se ha completado correctamente; un valor distinto de cero significa que ha habido un error. Si el valor devuelto es distinto de cero, puede llamar a `GetLastErrorString()` o `GetLastErrorType()` para obtener información adicional del error.

## ShowBusyWindow()

`void ShowBusyWindow(BOOL Mostrar)`

### Descripción

Esta función indica a QMF para Windows que muestre u oculte la ventana de proceso en curso. La ventana de proceso en curso es útil para proporcionar información de retorno al usuario y para permitir al usuario cancelar una acción pendiente de la base de datos. Esta función solamente funciona si ha llamado a `SetBusyWindowMode()` con una modalidad de `RSM_CLIENTCONTROLLED`. Si ha establecido una ventana padre llamando a `SetParent()`, la ventana de proceso en curso permanecerá fija en la ventana especificada.

### Parámetros

Nombre	Descripción
Mostrar	Un valor distinto de cero muestra la ventana de proceso en curso; cero oculta la ventana de proceso en curso. Si el valor es distinto de cero, la ventana de proceso en curso se visualizará hasta que llame a <code>ShowBusyWindow()</code> con <i>Mostrar</i> establecido en cero.

### Valor devuelto

Ninguno.

## StartBind()

`short StartBind(BSTR NombreColección, BSTR NombrePaquete, BSTR SímboloConsist)`

### Descripción

Esta función empieza el proceso de vinculación de un paquete en la base de datos.

### Parámetros

Nombre	Descripción
NombreColección	ID de colección deseada para el paquete.
NombrePaquete	Nombre deseado para el paquete.
SímboloConsist	Serie de 16 caracteres de longitud que contiene la representación hexadecimal de un símbolo de ocho bytes utilizado para asegurar la coherencia entre el paquete vinculado en la base de datos y una aplicación que utiliza dicho paquete. Al ejecutar una sección del paquete, deberá proporcionar este mismo valor.

### Valor devuelto

Cero indica que la función se ha completado correctamente; un valor distinto de cero significa que ha habido un error. Si el valor devuelto es distinto de cero,

puede llamar a `GetLastErrorString()` o `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` o `GetLastSQLState()` para obtener información adicional del error.

### **Temas relacionados**

`EndBind()`

`CancelBind()`



---

## Apéndice. Avisos

Esta publicación ha sido desarrollada según los productos y servicios que se ofrecen en los Estados Unidos. Es posible que en otros países IBM no ofrezca los productos, los servicios o las características que se describen en este documento. Consulte con el representante local de IBM para obtener información sobre los productos y servicios disponibles actualmente en su área. Las referencias hechas a productos, programas o servicios de IBM no pretenden afirmar ni implicar que sólo puedan utilizarse esos productos, programas o servicios de IBM. Puede utilizarse cualquier otro producto, programa o servicio funcionalmente equivalente que no infrinja los derechos de propiedad intelectual de IBM. No obstante, es responsabilidad del usuario la evaluación y la verificación del funcionamiento de cualquier producto, programa o servicio que no sea de IBM.

IBM puede tener patentes o solicitudes de patente pendientes que cubran algunos temas descritos en este documento. La adquisición de este documento no confiere ninguna licencia sobre dichas patentes. Puede enviar consultas sobre licencias, por escrito, a:

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10594-1785  
Estados Unidos

Para realizar consultas sobre licencias relacionadas con información del juego de caracteres de doble byte (DBCS), contacte con el Departamento de Propiedad Intelectual de IBM de su país, o bien envíe las consultas, por escrito, a:

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

**El párrafo siguiente no se aplica al Reino Unido ni a ningún otro país donde las disposiciones en él expuestas sean incompatibles con la legislación local:**  
INTERNATIONAL BUSINESS MACHINES CORPORATION PROPORCIONA ESTA PUBLICACIÓN "TAL CUAL" SIN GARANTÍAS DE NINGÚN TIPO, YA SEAN EXPLÍCITAS O IMPLÍCITAS, INCLUYENDO, PERO SIN LIMITARSE A, LAS GARANTÍAS IMPLÍCITAS DE COMERCIALIZACIÓN O IDONEIDAD PARA UNA FINALIDAD DETERMINADA. Algunas legislaciones no contemplan la exclusión de garantías, ni implícitas ni explícitas, por lo que puede haber usuarios a los que no les afecte dicha norma.

Esta información puede contener imprecisiones técnicas o errores tipográficos. Periódicamente se efectúan cambios en la información aquí contenida; dichos cambios se incorporarán en nuevas ediciones de la publicación. IBM se reserva el derecho a realizar, si lo considera oportuno y sin previo aviso, cualquier modificación en los productos o programas que se describen en el presente manual.

Cualquier referencia en esta publicación a sitios Web que no sean de IBM se facilita con fines meramente informativos y en ningún caso supone un aval de dichos

sitios Web. El contenido de estos sitios Web no forma parte del contenido de este producto de IBM y el uso de los mismos es responsabilidad exclusiva del usuario.

Los poseedores de la licencia de este programa que deseen obtener información sobre el mismo con el propósito de: (i) intercambiar información entre programas creados de forma independiente y otros programas (incluido éste) y (ii) compartir la información intercambiada, deben ponerse en contacto con:

IBM Corporation  
J74/G4  
555 Bailey Avenue  
P.O. Box 49023  
San Jose, CA 95161-9023  
Estados Unidos

La disponibilidad de esta información puede estar sujeta a determinados términos y condiciones y, en algunos casos, al pago de alguna tasa.

IBM proporciona el programa bajo licencia descrito en esta documentación y todo el material con licencia disponible para el mismo bajo los términos del Acuerdo del cliente de IBM, el Acuerdo internacional de programa bajo licencia de IBM, o cualquier otro acuerdo equivalente entre ambas partes.

Todos los datos sobre rendimiento que figuran en el presente manual se han obtenido en un entorno controlado. Por consiguiente, los resultados obtenidos en otros entornos operativos pueden variar sustancialmente. Es posible que algunas mediciones se hayan realizado en sistemas en desarrollo, por lo que no existe garantía alguna de que dichas mediciones sean las mismas en sistemas puestos a disposición del público. Además, puede que alguna medición se haya calculado a partir de una extrapolación. Los resultados reales pueden variar. Los usuarios de este documento deberán comprobar los datos aplicables a su entorno específico.

La información referente a productos que no son de IBM procede de los suministradores de dichos productos, de anuncios publicados sobre los mismos, o de otras fuentes disponibles públicamente. IBM no ha realizado pruebas con dichos productos y no está en posición de confirmar la precisión de su rendimiento, compatibilidad o cualquier otra reclamación relacionada con productos que no son de IBM. Las preguntas sobre las capacidades de estos productos deben dirigirse a sus suministradores respectivos.

Todas las declaraciones relacionadas con futuros planes o directrices de IBM están sujetas a cambios o retractaciones sin previo aviso, y únicamente representan meros propósitos u objetivos.

Todos los precios de IBM indicados son precios de minorista sugeridos por IBM, son actuales y están sujetos a cambios sin previo aviso. Los precios de concesionarios pueden ser distintos.

Esta información sirve únicamente con fines de planificación. Su contenido está sujeto a cambios que pueden producirse antes de que los productos descritos estén disponibles.

Este documento contiene ejemplos de datos e informes que se utilizan en las 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 parecido con los nombres y direcciones utilizados por una empresa comercial real es pura coincidencia.

## LICENCIA DE COPYRIGHT:

Este documento contiene programas de aplicación de ejemplo en el idioma de origen, cuya misión es ilustrar las técnicas de programación en distintas plataformas operativas. El usuario puede copiar, modificar y distribuir dichos programas de ejemplo en la forma que prefiera sin previo pago a IBM, con el objeto de desarrollar, utilizar, comercializar o distribuir los programas de aplicación de acuerdo con la interfaz de programación de aplicaciones diseñada para la plataforma operativa para la que se han escrito dichos programas de ejemplo. Estos ejemplos no han sido sometidos a pruebas exhaustivas bajo todas las condiciones posibles. Por consiguiente, IBM no puede garantizar ni confirmar la fiabilidad, utilidad o funcionalidad de estos programas.

Si está viendo este documento en copia software, puede que no aparezcan las fotografías ni las ilustraciones en color.

---

## Marcas registradas

Los términos que siguen son marcas registradas de IBM Corporation en los Estados Unidos y/o en otros países:

ACF/VTAM	IBMLink
Advanced Peer-to-Peer Networking	IMS
AIX	Language Environment
AIX/6000	MVS/ESA
AS/400	MVS/XA
CICS	OfficeVision/VM
CICS/ESA	OS/2
CICS/MVS	OS/390
CICS/VSE	PL/I
COBOL/370	PROFS
DATABASE 2	QMF
DataJoinerDB2	RACF
DB2 Universal Database	S/390
Distributed Relational Database Architecture	SQL/DS
DRDA	Virtual Machine/Enterprise Systems Architecture
DXTGDDM	Visual Basic
IBM	VM/XA
	VM/ESA
	VSE/ESAVTAM

Java, o todas las marcas registradas y logotipos basados en Java, y Solaris son marcas registradas de Sun Microsystems, Inc. en los Estados Unidos, en otros países, o en ambos.

Lotus y 1-2-3 son marcas registradas de Lotus Development Corporation en los Estados Unidos, en otros países, o en ambos.

Microsoft, Windows y Windows NT son marcas registradas de Microsoft Corporation.

Otros nombres de empresas, productos y servicios, que pueden estar marcados con dos asteriscos (\*\*), pueden ser marcas registradas o de servicio de otras compañías.



---

# Índice

## A

abrir

- archivos de consultas solicitud 19
- archivos de SQL guardados 11
- consultas de SQL en el servidor de bases de datos 11
- consultas solicitud en el servidor de bases de datos 19
- formatos guardados en el servidor de bases de datos 29
- procedimientos en la base de datos 33

AddDecimalHostVariable() 62

AddHostVariable() 63

adición

- condiciones de clasificación 15
- condiciones de fila 16

adición de una fila

- editor de tablas 48

agrupación de resultados de consulta 23, 41

aplicaciones de ejemplo 54

aplicar formato a resultados de consulta numéricos 23

archivos

- exportación de datos 51

archivos de trabajos, creación 39

Avisos 125

## B

barra de herramientas

- adición de botones 5
- eliminación de botones 6
- mover botones 5
- personalización 5

base de datos

- seguridad 2

BindDecimalHostVariable() 64

BindHostVariable() 64

BindSection() 65

Bloqueo de llamadas 61

botones de acción

- consulta solicitud 13

búsqueda

- editor de tablas 47

## C

cálculos

- formatos 25

cambio de contraseñas 3

cambio de tamaño de las columnas y filas 21

cambio de una fila

- editor de tablas 48

CancelBind() 66

clasificación de resultados de consulta 22, 39

ClearList() 67

Close() 67

columnas

- añadir a consultas solicitud 14
- formatos 25

columnas, reordenación 22, 40

columnas, selección 21

Commit() 68

cómo añadir objetos a las listas 36

cómo aplicar formato a resultados de consulta numéricos 40

cómo guardar resultados de consulta 23, 41

cómo guardar resultados de consulta en archivos 24, 41

cómo resumir resultados de consulta 23, 41

CompleteQuery() 68

condiciones

- formatos 25

condiciones de clasificación

- adición 15
- utilización 15

condiciones de fila

- adición 16
- contiene 16
- empieza por 16
- entre 16
- igual a 16
- mayor que 16
- mayor que o igual a 16
- menor que 16
- menor que o igual a 16
- NULO 16
- termina por 16
- utilización 16

condiciones de unión

- creación en consultas solicitud 17

Conexión a la base de datos 62

Consulta de la API 62

consulta solicitud

- botones de acción 13

consultas

- crear complejas 14

consultas complejas

- creación 14

consultas de SQL

- abrir archivos guardados 11
- abrir un documento nuevo 7
- ejecución en el servidor de bases de datos 7
- guardar en archivos 11
- guardar en el servidor de bases de datos 11
- impresión 12
- presentación preliminar 12

consultas estáticas

- creación 43
- ejecución 45
- utilización de variables de sustitución 43

consultas solicitud

- abrir archivos guardados 19

consultas solicitud (*continuación*)

- añadir columnas 14
- añadir tablas 14
- conversión a SQL 18
- creación 13
- creación de condiciones de unión 17
- ejecución 14
- guardar como archivos 19
- guardar en el servidor de bases de datos 19
- utilización de múltiples tablas 17
- utilización de SQL 18
- utilización de variables de sustitución 18
- visualización de SQL 18

contiene (condición de fila) 16

contraseñas

- corrección 3

conversión del formato en un formato 23, 41

CopyToClipboard() 68

Correo cd Internet 53

cortes

- formatos 25

creación

- consultas estáticas 43

creación de archivos de trabajos 39

creación de un procedimiento lineal 31

cuentas 3

## CH

ChangePassword() 66

## D

DB2 Forms 49

DeleteQMFOject() 69

describir consultas

- creación 9

describir objeto 36

detalles

- formatos 25

## E

editar objeto 36

editor de tablas 47

- adición de una fila 48
- búsqueda de filas 47
- cambio de una fila 48

ejecución

- consulta de SQL en un servidor de bases de datos 7
- consultas estáticas 45
- consultas solicitud 14

ejecutar objeto 36

eliminación de objetos de las listas 36

empieza por (condición de fila) 16

EndBind() 70

entre (condición de fila) 16

- Enviar a 53
- es (operador de condición de fila) 16
- Excel
  - macro automática 53
- Execute() 70
- ExecuteEx() 70
- ExecuteStoredProcedure() 71
- ExecuteStoredProcedureEx() 72
- Export() 73
- exportación
  - informes 30
- exportación de datos
  - a archivos 51
  - a otras tablas 52
  - en un servidor de bases de datos 52
- ExportForm() 75
- ExportReport() 76

## F

- FastSaveData() 77
- FetchNextRow() 78
- FetchNextRowEx() 79
- FetchNextRows() 79
- FetchNextRowsEx() 80
- filas, selección 21
- final
  - formatos 25
- FlushQMFCache() 81
- formato
  - principal 25
- formatos
  - abrir un archivo guardado 28, 32
  - cálculos 25
  - columnas 25
  - condiciones 25
  - cortes 25
  - detalles 25
  - final 25
  - guardar en archivos 28, 32
  - guardar en el servidor de bases de datos 28, 32
  - HTML 25
  - opciones 25
  - página 25
  - producir un informe 26
- fuentes
  - visualización de la consulta 8
  - visualización de resultados 22, 40

## G

- GetColumnCount() 81
- GetColumnDataValue() 82
- GetColumnHeader() 82
- GetColumnHeaderEx() 82
- GetColumnHeadings() 83
- GetColumnValue() 84
- GetColumnValueEx() 84
- GetDefaultServerName() 85
- GetGlobalVariable() 85
- GetHostVariableNames() 85
- GetHostVariableTypes() 86
- GetLastErrorString() 86
- GetLastErrorType() 87
- GetLastSQLCode() 88
- GetLastSQLError() 88

- GetLastSQLState() 89
- GetOption() 89
- GetOptionEx() 91
- GetProcText() 91
- GetProcVariables() 91
- GetQMFOBJECTInfo() 92
- GetQMFOBJECTInfoEx() 94
- GetQMFOBJECTList() 95
- GetQMFOBJECTListEx() 96
- GetQMFProcText() 97
- GetQMFQueryText() 97
- GetQueryText() 98
- GetQueryVerb() 98
- GetResourceLimit() 98
- GetResourceLimitEx() 102
- GetRowCount() 102
- GetServerList() 103
- GetServerListEx() 104
- GetStoredProcedureResultSets() 104
- GetVariables() 105
- GetVariablesEx() 105
- gobierno 4
- guardar
  - consultas de SQL en archivos 11
  - consultas de SQL en el servidor de bases de datos 11
  - consultas solicitud como archivos 19
  - consultas solicitud en el servidor de bases de datos 19
  - formatos en archivos 28, 32
  - formatos en el servidor de bases de datos 28, 32

## H

- HTML
  - formatos 25

## I

- igual a (condición de fila) 16
- impresión
  - consultas de SQL 12
  - informes 29
  - procedimiento 33
- impresión de resultados de consulta 24, 42
- informes
  - exportación 30
  - impresión 29
  - presentación preliminar 26
  - producir un informe utilizando formatos 26
- inicio de sesión 2
- InitializeProc() 106
- InitializeQuery() 106
- InitializeServer() 107
- InitializeStaticQuery() 108
- IsStatic() 108

## L

- listado
  - objetos 35
- listas
  - abrir archivos guardados 37
  - listas, añadir objetos 36

- listas, eliminación de objetos 36

## M

- macro automática
  - Excel 53
- mayor que (condición de fila) 16
- mayor que o igual a (condición de fila) 16
- menor que (condición de fila) 16
- menor que o igual a (condición de fila) 16
- múltiples consultas
  - visualización simultánea 8
- múltiples documentos de consulta 8
- múltiples tablas
  - en consultas solicitud 17

## N

- no es (operador de condición de fila) 16
- nueva
  - consulta de SQL 7
  - consultas solicitud 13
  - describir consultas 9
- NULO (condición de fila) 16

## O

- objetos
  - listado 35
- opciones
  - formatos 25
- Open() 109
- operador de condición de fila
  - Es 16
  - No es 16

## P

- página
  - formatos 25
- Prepare() 110
- presentación preliminar
  - consultas impresas 12
  - consultas solicitud 20
  - informes 26
  - procedimiento impreso 33
- presentación preliminar de resultados de consulta 24, 42
- principal
  - formatos 25
- PrintReport() 110
- procedimiento
  - impresión 33
- procedimiento con lógica 31
- procedimiento REXX 31

## R

- ReinitializeServer() 110
- reordenación de columnas 22, 40
- resultados de consulta, agrupación 23, 41
- resultados de consulta, aplicar formato 23, 40

- resultados de consulta, clasificación 22, 39
- resultados de consulta, guardar 23, 41
- resultados de consulta, guardar en archivos 24, 41
- resultados de consulta, impresión 24, 42
- resultados de consulta, presentación preliminar 24, 42
- resultados de consulta, resumir 23, 41
- Rollback() 111
- RunProc() 111

## S

- SaveData() 111
- SaveQMFPProc() 113
- SaveQMFPQuery() 114
- Selección de columnas y filas 21
- servidor
  - establecimiento 1
- servidor de bases de datos
  - exportación de datos 52
- SetBindOption() 114
- SetBindOwner() 116
- SetBusyWindowButton() 116
- SetBusyWindowMessage() 117
- SetBusyWindowMode() 117
- SetBusyWindowTitle() 118
- SetGlobalVariable() 118
- SetHostVariable() 119
- SetOption() 119
- SetParent() 120
- SetProcVariable() 121
- SetVariable() 121
- ShowBusyWindow() 122
- SQL
  - utilización en consultas solicitud 18
- StartBind() 122

## T

- tablas
  - añadir a consultas solicitud 14
  - exportación de datos 52
- termina por 16

## V

- variables de sustitución
  - ejecución de consultas de SQL 10
  - en consultas de SQL 10
  - sustituir por variables del sistema principal 43
  - utilización en consultas solicitud 18
  - utilizar en consultas estáticas 43
- variables del sistema principal
  - utilizar en consultas estáticas 43
- vista
  - resultados 7
  - SQL 7
- vista de resultados 7
- visualización
  - SQL en consultas solicitud 18
- visualizar objetos 36



---

# Hoja de Comentarios

Query Management Facility  
Iniciación a QMF para Windows  
Versión 7

Número de Publicación SC10-3533-00

Por favor, sírvase facilitarnos su opinión sobre esta publicación, tanto a nivel general (organización, contenido, utilidad, facilidad de lectura,...) como a nivel específico (errores u omisiones concretos). Tenga en cuenta que los comentarios que nos envíe deben estar relacionados exclusivamente con la información contenida en este manual y a la forma de presentación de ésta.

Para realizar consultas técnicas o solicitar información acerca de productos y precios, por favor diríjase a su sucursal de IBM, business partner de IBM o concesionario autorizado.

Para preguntas de tipo general, llame a "IBM Responde" (número de teléfono 901 300 000).

Al enviar comentarios a IBM, se garantiza a IBM el derecho no exclusivo de utilizar o distribuir dichos comentarios en la forma que considere apropiada sin incurrir por ello en ninguna obligación con el remitente.

Comentarios:

Gracias por su colaboración.

Para enviar sus comentarios:

- Envíelos por correo a la dirección indicada en el reverso.
- Envíelos por correo electrónico a: <http://www.dss.ibm.com>

Si desea obtener respuesta de IBM, rellene la información siguiente:

Nombre

Dirección

Compañía

Número de teléfono

Dirección de e-mail

IBM S.A.  
National Language Solutions Center  
Av. Diagonal 571, Edif. "L'Illa"  
08029 Barcelona  
España





Número de Archivo:  
Número de Programa: 5675-DB2  
5697-F42  
5697-G24  
5697-G22  
5648-D35  
5697-G23

SC10-3533-00



Spine information:



QMF

Iniciación a QMF para Windows

Versión 7