

Query Management Facility



# Informações Iniciais do QMF para Windows

*Versão 7*



Query Management Facility



# Informações Iniciais do QMF para Windows

*Versão 7*

**Nota!**

Antes de utilizar estas informações e o produto a que elas dão suporte, não deixe de ler as informações gerais em “Apêndice. Avisos” na página 149.

**Primeira Edição (Setembro de 2000)**

Esta edição aplica-se ao Query Management Facility para Windows, um recurso da Versão 7 do DB2 Universal Database Server para OS/390 (DB2 UDB para OS/390), 5675-DB2 e ao Query Management Facility, um recurso da Versão 7 do DATABASE 2 Server para VM e VSE (DB2 para VM e VSE), 5697-F42, do Query Management Facility para Windows para AS/400, 5697-G24, do Query Management Facility para Windows para DB2 Workstation Databases, 5697-G22, do DB2 Warehouse Manager, 5648-D35 e do DB2 Warehouse Manager para AS/400, 5697-G23 e a todos os releases e modificações subseqüentes, até que seja indicado de outra forma em novas edições.

Esta edição substitui e torna obsoleta a edição anterior, SC26-9582-02. Alterações editoriais que não tem significado técnico não serão notificadas.

© Copyright International Business Machines Corporation 1997, 2000. Todos os direitos reservados.

# Índice

<b>A Biblioteca do QMF</b> . . . . .	<b>vii</b>	Abertura de consultas SQL salvas no servidor de banco de dados . . . . .	14
<b>Capítulo 1. Introdução</b> . . . . .	<b>1</b>	Impressão de consultas SQL . . . . .	15
Servidores de Banco de Dados . . . . .	1	Visualização de uma consulta . . . . .	15
Família DB2 de bancos de dados . . . . .	1	Impressão de consultas SQL . . . . .	15
Nome de usuário vs. nome técnico . . . . .	1		
Definição do nome do servidor . . . . .	1	<b>Capítulo 3. Operação com Consultas Orientadas</b> . . . . .	<b>17</b>
Segurança do banco de dados . . . . .	2	Construção de Consultas Simples . . . . .	17
Início de Sessão . . . . .	2	Abertura de novas consultas orientadas . . . . .	17
Correção de senhas . . . . .	3	Botões de ação da consulta orientada . . . . .	17
Alteração de senhas . . . . .	3	Inclusão de tabelas e consultas orientadas . . . . .	18
Especificação de cadeias de conta . . . . .	4	Execução de consultas orientadas . . . . .	18
Administração . . . . .	4	Construção de Consultas Complexas . . . . .	19
Exibição dos limites de recursos . . . . .	4	Inclusão de colunas em consultas orientadas . . . . .	19
Definição do seu próprio limite de linhas . . . . .	5	Uso de condições de ordenação . . . . .	20
Personalização da barra de ferramentas . . . . .	6	Inclusão de condições de ordenação . . . . .	20
Adição de botões na barra de ferramentas . . . . .	6	Uso de condições de linha . . . . .	20
Movimentação de botões da barra de ferramentas . . . . .	6	Inclusão de condições de linha . . . . .	21
Remoção de botões da barra de ferramentas . . . . .	7	Uso de múltiplas tabelas em consultas orientadas . . . . .	22
		Criação de condições de junção de consulta orientada . . . . .	22
<b>Capítulo 2. Operação com Consultas SQL</b> . . . . .	<b>9</b>	Consultas Orientadas e SQL . . . . .	22
Consultas SQL . . . . .	9	Exibição de SQL para consultas orientadas . . . . .	23
Criação de novas consultas SQL . . . . .	9	Conversão de consultas orientadas em SQL . . . . .	23
Execução de consultas SQL em um servidor de banco de dados . . . . .	9	Uso de Variáveis de Substituição nas Consultas Orientadas . . . . .	23
Alternância entre a Exibição de Resultados e a Exibição de SQL . . . . .	9	Salvamento de Consultas Orientadas . . . . .	23
Operação com fontes . . . . .	10	Salvamento de consultas orientadas em arquivos . . . . .	23
Seleção da fonte de exibição da consulta . . . . .	10	Abertura de arquivos salvos de consulta orientada . . . . .	24
Múltiplas consultas . . . . .	10	Salvamento de consultas orientadas no servidor de banco de dados . . . . .	24
Exibição de múltiplas consultas simultaneamente . . . . .	10	Abertura de consultas orientadas salvas no servidor de banco de dados . . . . .	24
Desenho de consultas . . . . .	11	Impressão de Consultas Orientadas . . . . .	25
Criação de novas consultas SQL . . . . .	11	Visualização de consultas orientadas . . . . .	25
Variáveis de substituição em consultas SQL . . . . .	12		
Execução de consultas SQL com variáveis de substituição . . . . .	12	<b>Capítulo 4. Operação com Resultados de Consultas</b> . . . . .	<b>27</b>
Salvamento e abertura de consultas SQL . . . . .	13	Classificação e dimensionamento de resultados de consultas . . . . .	27
Salvamento de consultas SQL em arquivos . . . . .	13		
Aberturas de arquivos de consulta SQL salvos . . . . .	13		
Salvamento de consultas SQL no servidor de banco de dados . . . . .	14		

Seleção de colunas e linhas . . . . .	27
Redimensionamento de colunas e linhas . . . . .	27
Ajuste automático de colunas e linhas . . . . .	28
Classificação de resultados da consulta . . . . .	28
Reordenação de colunas . . . . .	28
Formatação de resultados da consulta . . . . .	28
Seleção da fonte de exibição dos resultados da consulta . . . . .	29
Formatação de resultados de consulta numéricos. . . . .	29
Conversão da formatação de resultados da consulta em um formulário . . . . .	29
Agrupamento e agregação de resultados da consulta . . . . .	29
Agrupamento de resultados da consulta . . . . .	29
Resumo de resultados da consulta . . . . .	30
Salvamento de resultados da consulta e formatação . . . . .	30
Salvamento dos resultados da consulta como uma tabela . . . . .	30
Salvamento de resultados da consulta em arquivos . . . . .	30
Impressão de resultados da consulta . . . . .	30
Visualização de resultados da consulta . . . . .	30
Impressão de resultados da consulta . . . . .	31
<b>Capítulo 5. Operação com Relatórios . . . . .</b>	<b>33</b>
Formulários . . . . .	33
Compreensão de Formulários . . . . .	33
Produção de um relatório usando um formulário . . . . .	34
Edição de um formulário. . . . .	34
Criação de Formulário . . . . .	35
Passo 1: Criar um formulário . . . . .	35
Passo 2: Alterar a ordem das colunas. . . . .	35
Passo 3: Alterar os títulos das colunas . . . . .	35
Passo 4: Alterar o formato da coluna. . . . .	36
Passo 5: Incluir informações de resumo . . . . .	36
Passo 6: Incluir cabeçalhos e rodapés de páginas . . . . .	36
Salvamento de formulários . . . . .	37
Salvamento de um formulário em um arquivo . . . . .	37
Abertura de arquivos de formulário salvos . . . . .	37
Salvamento de formulários no servidor de banco de dados . . . . .	37
Abertura de formulários salvos no servidor de banco de dados . . . . .	38
Impressão de relatórios . . . . .	38
Exportação de relatórios . . . . .	39

<b>Capítulo 6. Operação com Procedimentos . . . . .</b>	<b>41</b>
Execução de Procedimentos . . . . .	41
Criação de um novo procedimento linear . . . . .	41
Criação de um novo procedimento com lógica . . . . .	41
Execução de um procedimento em um servidor de banco de dados. . . . .	41
Salvamento de Procedimentos . . . . .	42
Salvamento de procedimentos em arquivos . . . . .	42
Abertura de um arquivo de procedimento salvo . . . . .	42
Salvamento de um procedimento no servidor de banco de dados. . . . .	42
Abertura de procedimentos salvos no servidor de banco de dados. . . . .	43
Impressão de Procedimentos . . . . .	43
Visualização de um procedimento. . . . .	43
Impressão de um procedimento . . . . .	44

<b>Capítulo 7. Operação com Listas . . . . .</b>	<b>45</b>
Objetos. . . . .	45
Lista de objetos . . . . .	45
Comandos da Janela Lista . . . . .	46
Criação de Listas . . . . .	46
Inclusão de objetos em listas . . . . .	46
Remoção de objetos das listas . . . . .	47
Salvamento de listas em arquivos . . . . .	47
Abertura de arquivos de lista salvos . . . . .	47

<b>Capítulo 8. Operação com Arquivos de Job . . . . .</b>	<b>49</b>
Arquivos de job. . . . .	49
Criação de arquivos de job . . . . .	49
Execução de arquivos de job . . . . .	49
Ajuste automático de colunas e linhas . . . . .	49
Classificação de resultados da consulta . . . . .	50
Reordenação de colunas . . . . .	50
Formatação de resultados da consulta . . . . .	50
Seleção da fonte de exibição dos resultados da consulta . . . . .	50
Formatação de resultados de consulta numéricos. . . . .	51
Conversão da formatação de resultados da consulta em um formulário . . . . .	51
Agrupamento e agregação de resultados da consulta . . . . .	51
Agrupamento de resultados da consulta . . . . .	51
Resumo de resultados da consulta . . . . .	51
Salvamento de resultados da consulta e formatação . . . . .	51

Salvamento dos resultados da consulta como uma tabela . . . . .	52
Salvamento de resultados da consulta em arquivos . . . . .	52
Impressão de resultados da consulta . . . . .	52
Visualização de resultados da consulta . . . . .	52
Impressão de resultados da consulta . . . . .	52

## Capítulo 9. Operação com Consultas

<b>Estáticas . . . . .</b>	<b>55</b>
Consultas Estáticas . . . . .	55
Criação de consultas estáticas . . . . .	55
Substituição de variáveis de substituição com variáveis do host. . . . .	56
Execução de uma consulta estática . . . . .	57

## Capítulo 10. Operação com o Editor de Tabelas

<b>Editor de tabela . . . . .</b>	<b>59</b>
Pesquisa de linhas usando o editor de tabela . . . . .	59
Inclusão de uma linha . . . . .	60
Alteração de linha . . . . .	60
Eliminação de uma linha. . . . .	60
Edição de Tabelas a Partir da Exibição dos Resultados da Consulta . . . . .	61
Eliminação de uma linha da exibição dos resultados da consulta . . . . .	61
Atualização de colunas a partir da exibição dos resultados da consulta . . . . .	61
DB2 Forms . . . . .	61

## Capítulo 11. Distribuição de Dados

<b>Exportação de dados . . . . .</b>	<b>63</b>
Exportação de dados para arquivos . . . . .	63
Importação de dados . . . . .	64
Salvamento de dados em um servidor de banco de dados . . . . .	65
Uso do comando Enviar Para . . . . .	65
Uso do Complemento Microsoft Excel . . . . .	66
Uso das Aplicações de Exemplo . . . . .	66

## Capítulo 12. Uso da Central de Relatórios do QMF

<b>Introdução à Central de Relatórios do QMF</b>	<b>67</b>
Janela da Central de Relatórios do QMF	67
Conexão ao Servidor . . . . .	68
Operação com Relatórios e Objetos . . . . .	69
Execução de Relatórios . . . . .	70
Operação com Pastas e Favoritos . . . . .	70

Adição de Relatórios a Favoritos . . . . .	71
--	----

## Capítulo 13. Uso da API do QMF para

### Windows. . . . . 73

Controle do QMF para Windows pela API . . . . .	73
Bloqueio de chamadas . . . . .	74
Conexão ao banco de dados. . . . .	74
Referência da API . . . . .	75
AddDecimalHostVariable() . . . . .	75
AddHostVariable() . . . . .	76
BindDecimalHostVariable() . . . . .	77
BindHostVariable() . . . . .	77
BindSection(). . . . .	78
CancelBind() . . . . .	79
ChangePassword() . . . . .	79
ClearList(). . . . .	80
Close() . . . . .	80
Commit() . . . . .	81
CompleteQuery() . . . . .	81
CopyToClipboard(). . . . .	82
DeleteQMFObject(). . . . .	83
EndBind(). . . . .	83
Execute() . . . . .	84
ExecuteEx() . . . . .	84
ExecuteStored Procedure() . . . . .	85
ExecuteStored ProcedureEx() . . . . .	86
Export() . . . . .	88
ExportForm() . . . . .	90
ExportReport() . . . . .	91
FastSaveData() . . . . .	92
FetchNextRow(). . . . .	93
FetchNextRowEx() . . . . .	94
FetchNextRows() . . . . .	95
FetchNextRowsEx() . . . . .	97
FlushQMFCache() . . . . .	97
GetColumnCount(). . . . .	98
GetColumnDataValue() . . . . .	98
GetColumnHeader() . . . . .	98
GetColumnHeaderEx() . . . . .	99
GetColumnHeadings() . . . . .	100
GetColumnValue() . . . . .	101
GetColumnValueEx(). . . . .	101
GetDefaultServerName() . . . . .	102
GetGlobalVariable() . . . . .	102
GetHostVariableNames() . . . . .	102
GetHostVariableTypes(). . . . .	103
GetLastErrorString() . . . . .	103
GetLastErrorType() . . . . .	104
GetLastSQLCode() . . . . .	105
GetLastSQLException() . . . . .	106

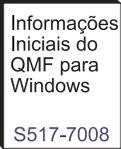
GetLastSQLState(). . . . .	107	PrintReport() . . . . .	132
GetOption(). . . . .	107	ReinitializeServer() . . . . .	132
GetOptionEx() . . . . .	109	Rollback() . . . . .	133
GetProcText() . . . . .	109	RunProc() . . . . .	133
GetProcVariables() . . . . .	110	SaveData() . . . . .	134
GetQMFObjectInfo() . . . . .	111	SaveQMFPProc() . . . . .	136
GetQMFObjectInfoEx() . . . . .	113	SaveQMFPQuery() . . . . .	137
GetQMFObjectList() . . . . .	114	SetBindOption() . . . . .	137
GetQMFObjectListEx() . . . . .	115	SetBindOwner() . . . . .	139
GetQMFPProcText() . . . . .	116	SetBusyWindowButton() . . . . .	140
GetQMFPQueryText() . . . . .	117	SetBusyWindowMessage() . . . . .	140
GetQueryText(). . . . .	117	SetBusyWindowMode() . . . . .	141
GetQueryVerb() . . . . .	118	SetBusyWindowTitle() . . . . .	142
GetResourceLimit() . . . . .	118	SetGlobalVariable() . . . . .	142
GetResourceLimitEx() . . . . .	122	SetHostVariable() . . . . .	143
GetRowCount() . . . . .	123	SetOption() . . . . .	143
GetServerList(). . . . .	123	SetParent() . . . . .	145
GetServerListEx() . . . . .	124	SetProcVariable() . . . . .	145
GetStoredProcedureResultSets() . . . . .	125	SetVariable() . . . . .	146
GetVariables() . . . . .	126	ShowBusyWindow() . . . . .	146
GetVariablesEx() . . . . .	126	StartBind() . . . . .	147
InitializeProc() . . . . .	127		
InitializeQuery() . . . . .	128	<b>Apêndice. Avisos . . . . .</b>	<b>149</b>
InitializeServer() . . . . .	128	Marcas . . . . .	152
InitializeStaticQuery() . . . . .	130		
IsStatic() . . . . .	130	<b>Índice Remissivo . . . . .</b>	<b>153</b>
Open() . . . . .	130		
Prepare(). . . . .	131		

---

## A Biblioteca do QMF

Você pode solicitar manuais por meio de um representante da IBM ou pelo telefone 0-800-784-262 no Brasil.

---

Avaliação				
Instalação, planejamento, administração e diagnóstico				
				
Uso				
Programação de aplicativos				
Bibliotecas online	 SK2T-0730 OS/390, VM, & VSE	 Apenas SK2T-6700 OS/390	 Apenas SK2T-2067 VM	 Apenas SK2T-0060 VSE

---



---

# Capítulo 1. Introdução

Este capítulo apresenta uma visão geral do QMF para Windows e fornece explicações sobre algumas tarefas básicas para informações iniciais do QMF para Windows.

---

## Servidores de Banco de Dados

As consultas, formulários, procedimentos e tabelas são executados e salvos em um servidor de banco de dados.

### Família DB2 de bancos de dados

O QMF para Windows pode conectar-se a uma grande variedade de bancos de dados DB2.

- DB2 UDB for OS/390, DB2 for OS/390 e DB2 for MVS
- DB2 Server for VSE & VM e SQLDS
- DB2 Universal Database e DB2 Common Server
- DB2 Parallel Edition
- DataJoiner

Sua licença do QMF para Windows determina em quais produtos da família DB2 você pode efetuar a instalação e conectar-se, com sua cópia do QMF para Windows.

### Nome de usuário vs. nome técnico

Versões e tipos diferentes de DB2 referem-se a um banco de dados através de um nome RDB, nome de localização ou outro nome técnico.

Com o QMF para Windows, seu administrador pode atribuir a um banco de dados um nome de usuário que possa ser lembrado facilmente, como por exemplo, Banco de Dados de Compras ao invés de DB2P\_01\_PURCH.

O QMF para Windows refere-se a um servidor de banco de dados ou a um banco de dados DB2 como um "servidor."

### Definição do nome do servidor

Antes de você poder consultar um banco de dados, o QMF para Windows precisa saber onde o banco de dados está armazenado.

1. A partir do menu **Arquivo**, selecione **Nova Consulta SQL**. Um novo documento de consulta SQL é aberto.

2. A partir do menu **Consulta**, selecione **Definir Servidor**. A caixa de diálogo Definir Servidor aparece.



3. Da lista de servidores disponíveis, selecione aquele que deseja consultar e clique em **OK**. Quando a próxima sessão do QMF para Windows for iniciada, o QMF para Windows reconecta-se automaticamente ao mesmo servidor.

---

## Segurança do banco de dados

Você deve fornecer uma ID de usuário e senha antes de poder conectar-se a um servidor.

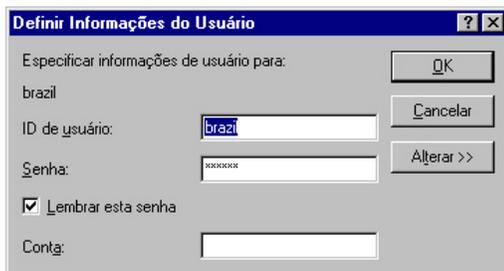
### Início de Sessão

É preciso especificar uma ID de usuário e uma senha que sejam válidas para o servidor de banco de dados que você está tentando acessar. A ID de usuário e senha do servidor de banco de dados não são necessariamente as mesmas que a ID de usuário e senha locais ou de rede.

Caso esteja executando o Windows, você terá a opção de conservar senhas do servidor entre sessões do QMF para Windows. Caso esteja no momento com sessão iniciada no Windows, a caixa de diálogo Definir Informações do Usuário mostrará uma caixa de opções adicional chamado **Lembrar esta senha**. Caso assinale este quadro, a senha fornecida para aquele servidor será armazenada na lista de senhas do Windows. Quando você estiver com sessão iniciada no Windows, o QMF para Windows pode recuperar automaticamente aquela senha de forma que não lhe seja solicitado. Se você não estiver com sessão iniciada ao executar o QMF para Windows ou se você estiver com sessão iniciada como outro usuário, o QMF para Windows solicita uma ID de usuário e uma senha.

**Nota:** Se optar por salvar uma senha, qualquer pessoa que possa iniciar sessão na sua conta do Windows poderá acessar os seus servidores de banco de dados com sua ID de usuário e senha (do servidor).

1. A partir do menu **Consulta**, selecione **Definir Informações do Usuário**. A caixa de diálogo Definir Informações do Usuário aparece.

A caixa de diálogo 'Definir Informações do Usuário' possui o seguinte conteúdo: no topo, o título 'Definir Informações do Usuário' e botões de ajuda e fechar; abaixo, o texto 'Especificar informações de usuário para:' seguido de um botão 'OK'; o campo 'brazil' para o nome do usuário; o campo 'ID de usuário:' com o valor 'brazil' digitado; o campo 'Senha:' com caracteres ocultos por pontos; um botão 'Cancelar'; um botão 'Alterar >>'; uma caixa de seleção marcada com o texto 'Lembrar esta senha'; e o campo 'Conta:'.

2. Digite a sua ID de usuário e a senha nos campos adequados.

**Nota:** A ID de usuário e a senha consideram maiúsculas e minúsculas. Por exemplo, se sua ID de usuário ou senha forem em letras maiúsculas, é preciso digitá-las em letras maiúsculas. Alguns tipos de servidores de banco de dados são sensíveis à letra usada nas IDs de usuário e senhas, ao passo que outros não são.

3. Assinale **Lembrar esta senha** caso deseje salvar sua ID de usuário e senha.
4. Clique em **OK**. O QMF para Windows armazena essas informações na preparação do acesso ao servidor.

### Correção de senhas

Se você digitou uma senha incorreta, você pode corrigir o erro reabrindo a caixa de diálogo Definir Informações do Usuário.

1. A partir do menu **Consulta**, clique em **Definir Informações do Usuário**. A caixa de diálogo Definir Informações do Usuário aparece.
2. Digite novamente a senha e clique em **OK**. A senha é corrigida.

### Alteração de senhas

Sua senha pode ser alterada no servidor de banco de dados a partir do QMF para Windows. Atualmente, este recurso é suportado apenas pelo DB2 for OS390 Versão 5 e posterior.

1. A partir do menu **Consulta**, selecione **Definir Informações do Usuário**. A caixa de diálogo Definir Informações do Usuário aparece.
2. Clique em **Alterar**. Os campos **Nova senha** e **Confirmar nova senha** aparecem.
3. Digite a nova senha nos campos **Nova senha** e **Confirmar nova senha** e clique em **OK**. A sua senha do servidor de banco de dados é alterada.

## Especificação de cadeias de conta

O servidor de banco de dados utiliza cadeias de conta para rastrear a utilização do sistema. Pergunte ao administrador do banco de dados se o seu sistema utiliza cadeias de conta.

1. A partir do menu **Consulta**, selecione **Definir Informações do Usuário**. A caixa de diálogo Definir Informações do Usuário aparece.
2. No campo **Conta**, digite a cadeia de conta que deseja utilizar e clique em **OK**. O QMF para Windows armazena as informações na preparação do acesso ao servidor.

---

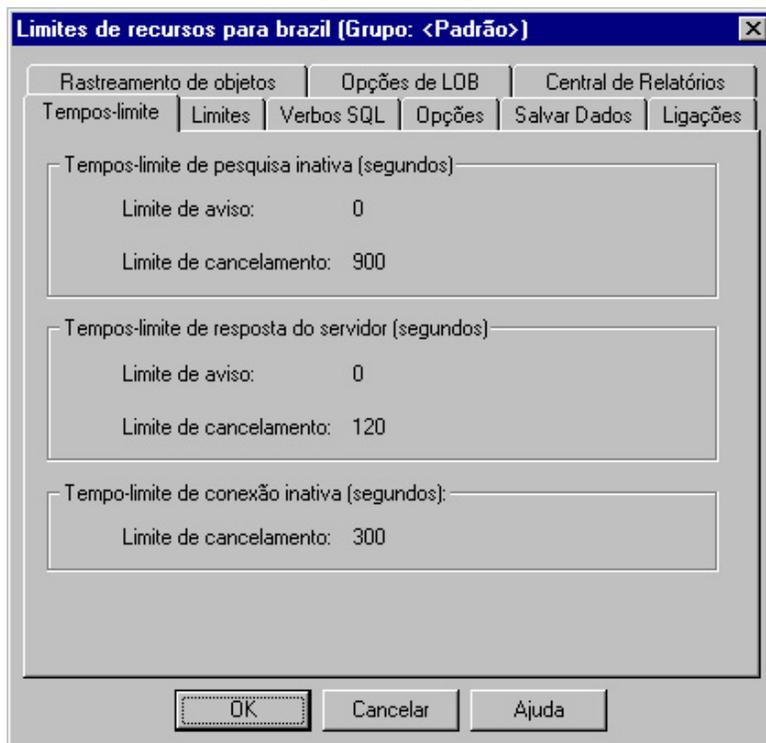
## Administração

O controlador do QMF para Windows está sempre em execução no plano de fundo, monitorando o uso dos recursos do banco de dados e do sistema. O controlador também limita o tipo e o tamanho das consultas que você pode executar.

### Exibição dos limites de recursos

No menu **Exibir**, selecione **Limites de Recursos**. A caixa de diálogo Limites de Recursos aparece. Todas as informações na caixa de diálogo Limites de Recursos são somente para leitura. O administrador do seu sistema define

esses limites.



Os tipos de limites e controles que podem estar em efeito são:

- Tempos-Limite
- Limites
- Verbos SQL
- Opções
- Salvar Dados
- Ligações
- Rastreamento de Objetos

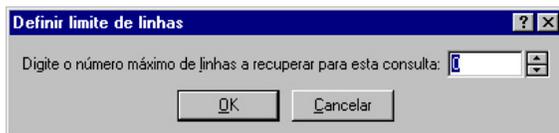
### **Definição do seu próprio limite de linhas**

Você pode especificar o número máximo de linhas a serem recuperadas para esta consulta. Ao atingir esse limite, o QMF para Windows cancela a consulta. O limite de linhas máximo autorizado, especificado no grupo de limites de recursos, tem precedência sobre esse parâmetro.

Digite 0 para não especificar limite nesse campo.

As linhas que o QMF para Windows já recuperou a mais em relação ao limite ficam retidas e disponíveis para visualização.

1. A partir do menu **Consulta**, clique em **Definir Limite de Linhas**. A caixa de diálogo Definir Limite de Linhas é aberta.



2. Digite o número máximo de linhas que deseja que a consulta retorne e clique em **OK**. O limite de linhas é aplicado na próxima vez que você executar a consulta.

---

## Personalização da barra de ferramentas

Você pode personalizar a barra de ferramentas a fim de exibir somente os botões que você deseja ver.

### Adição de botões na barra de ferramentas

Você tem a opção de incluir botões à barra de ferramentas existente do QMF para Windows. Estes botões representam funções que talvez não sejam necessárias a todos os usuários, mas que estão disponíveis para adição na barra de ferramentas.

1. Dê um clique duplo na área cinza que envolve a barra de ferramentas. A caixa de diálogo Personalizar barra de ferramentas aparece.



2. A partir da coluna **Botões disponíveis**, selecione o botão que deseja incluir e clique em **Adicionar**. O botão é incluído na barra de ferramentas.
3. Ao terminar de incluir botões, clique em **Fechar**. A caixa de diálogo é fechada e os novos botões são incluídos na barra de ferramentas.

### Movimentação de botões da barra de ferramentas

Você tem a opção de reorganizar os botões da barra de ferramentas do QMF para Windows.

1. Dê um clique duplo na área cinza que envolve a barra de ferramentas. A caixa de diálogo Personalizar barra de ferramentas aparece.
2. Na coluna **Botões disponíveis**, selecione o botão que você deseja mover.
3. Utilize os botões **Mover para cima** e **Mover para baixo** para mover o botão dentro da barra de ferramentas.
4. Ao terminar de mover botões, clique em **Fechar**. A caixa de diálogo é fechada e os botões aparecem nas suas novas localizações.

### **Remoção de botões da barra de ferramentas**

Você tem a opção de remover botões da barra de ferramentas do QMF para Windows.

1. Dê um clique duplo na área cinza que envolve a barra de ferramentas. A caixa de diálogo Personalizar barra de ferramentas aparece.
2. Na coluna **Botões disponíveis**, selecione o botão que deseja remover e clique em **Remover**. O botão é removido da barra de ferramentas.
3. Ao terminar de remover botões, clique em **Fechar**. A caixa de diálogo é fechada e a barra de ferramentas aparece com o botões removidos.



---

## Capítulo 2. Operação com Consultas SQL

Structured Query Language (SQL) é a mais básica interface entre um usuário e um banco de dados. As consultas são escritas em SQL e processadas pelo banco de dados. Os usuários podem escrever consultas do QMF para Windows em SQL ou criar consultas utilizando o método "apontar e clicar".

---

### Consultas SQL

As consultas em Structured Query Language exigem conhecimento de comandos e da sintaxe SQL. Os usuários que não estejam familiarizados com SQL devem tentar criar consultas orientadas.

#### Criação de novas consultas SQL

Clique o botão **Nova Consulta SQL** na barra de ferramentas.



Um novo documento de consulta é aberto.

#### Execução de consultas SQL em um servidor de banco de dados

1. Abra um novo documento de consulta e digite uma consulta, ou abra um arquivo de consulta existente, ou abra uma consulta do banco de dados.
2. Clique no botão **Executar Consulta** na barra de ferramentas.



A consulta é executada e os resultados são exibidos.

#### Alternância entre a Exibição de Resultados e a Exibição de SQL

Você pode examinar os resultados de uma consulta ou a própria instrução SQL.

A partir da exibição SQL de uma consulta que foi executada, clique no botão **Exibir Resultados** na barra de ferramentas.



Os resultados da consulta são exibidos.

-ou-

A partir da exibição Resultados de uma consulta, clique no botão **Exibir SQL**.



A instrução SQL é exibida.

---

## Operação com fontes

Você pode alterar a fonte utilizada para exibir consultas. A opção de fontes varia de acordo com o que está instalado em seu computador. Para obter mais informações sobre a inclusão de fontes, consulte o recurso de ajuda do sistema operacional.

**Nota:** Caso você salve uma consulta após selecionar uma nova fonte de exibição para a consulta, esta consulta sempre será exibida utilizando a fonte nova.

### Seleção da fonte de exibição da consulta

1. A partir da exibição SQL, clique em **Definir Fonte** no menu **Consulta**. A caixa de diálogo Fonte é aberta.
2. Selecione a fonte para exibição do texto da consulta e clique em **OK**. A consulta reaparece na nova fonte.

**Nota:** Clique em **Definir Como Padrão** para utilizar a fonte selecionada como fonte padrão para todas as consultas novas.

---

## Múltiplas consultas

Você pode ter mais do que um documento de consulta aberto ao mesmo tempo. Você também pode executar mais de uma consulta ao mesmo tempo. Este recurso pode ser utilizado para geração de múltiplos relatórios ou para cortar e colar o texto SQL de uma consulta para outra.

### Exibição de múltiplas consultas simultaneamente

1. Abra pelo menos dois documentos de consulta.
2. A partir do menu **Janela**, selecione um dos seguintes comandos:

Comando	Resultado
Cascata	Exibe consultas em séries escalonadas.
Paralelo Horizontal	Exibe as janelas de consulta escalonadas verticalmente.

As janelas de consulta são organizadas de acordo com a opção que você seleciona.

## Desenho de consultas

Use o comando Formular Consulta para criar novos documentos de consulta SQL. Você especifica um ou mais nomes de tabelas e o tipo de instrução SQL desejada e o QMF para Windows cria automaticamente uma instrução SQL que referencia os nomes e tipos de dados das colunas na tabela.

### Criação de novas consultas SQL

1. A partir do menu **Arquivo**, clique em **Formular Consulta**. A caixa de diálogo Formular Consulta é aberta.

2. Selecione o tipo de consulta que deseja criar:

Tipo de consulta	Resultado
Selecionar	Recupera linhas de uma ou mais tabelas.
Atualizar	Altera as informações em uma tabela.
Inserir	Inclui novas linhas em uma tabela.

3. Digite o proprietário e o nome da tabela a ser consultada.

**Nota:** Você pode usar padrões para selecionar nomes de tabelas a partir de uma lista de tabelas correspondentes.

- Use o caractere de porcentagem (%) para representar cadeia de qualquer tamanho que contenha qualquer caractere. Por exemplo, para listar todas as tabelas com um nome começando com a letra A, digite A%.
- Use o caractere de sublinhado (\_) para representar um caractere individual. Por exemplo, para listar todas as tabelas com um proprietário que possui a letra A na segunda posição, você digita \_A%.

Depois de fornecer um padrão, clique em **Incluir da Lista** e selecione uma tabela da lista resultante.

4. Digite um identificador exclusivo para tabela.
5. Clique em **Incluir**. A tabela é incluída na consulta.
6. Quando tiver incluído a tabela ou tabelas que deseja consultar, clique em **OK**. Uma consulta SQL para as tabelas selecionadas é criada e exibida.

---

## Variáveis de substituição em consultas SQL

Com as variáveis de substituição, é possível utilizar a mesma consulta para recuperar informações diferentes através do fornecimento de valores diferentes cada vez que uma consulta for executada. Para recuperar um conjunto de dados diferente, não é necessário reescrever a consulta. Ao invés disso, forneça valores diferentes para as variáveis de substituição na consulta quando você a executa.

Variável de substituição é o texto incluído em uma consulta. Ela deve começar com um caractere e comercial (&) e pode conter até 18 caracteres, que podem ser alfabéticos, numéricos ou um dos seguintes caracteres especiais: ^ ! \$ % & ' { } ? @ # % \ ou \_ . Os seguintes valores, por exemplo, são variáveis de substituição válidas:

```
&VARIABLE1
&DEPARTMENT_NUMBER
```

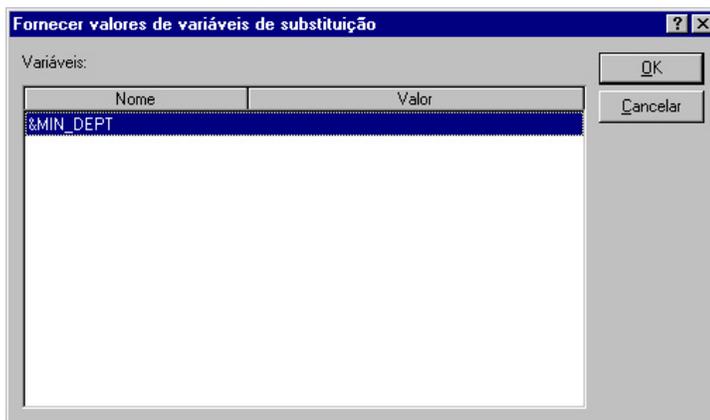
Uma variável de substituição pode aparecer em qualquer ponto de uma consulta e seu valor pode ser qualquer coisa que você escreva em uma consulta (exceto comentário). Por exemplo, a variável de substituição pode ser utilizada no lugar de um nome de coluna, de condições de pesquisa, de uma subconsulta ou de qualquer valor específico.

### Execução de consultas SQL com variáveis de substituição

1. Abra um novo documento de consulta e digite esta instrução SQL:  

```
SELECT * FROM Q.STAFF WHERE DEPT >= &MIN_DEPT
```

2. Execute a consulta. A caixa de diálogo Fornecer Valores de Variáveis de Substituição é aberta.



3. No campo **Valor**, digite um valor de 50 e clique em **OK**. A consulta executa e exibe os resultados de consulta.

Tente experimentar com as variáveis de substituição substituindo valores nas cláusulas SELECT e FROM. Veja quais resultados a sua consulta retorna com as diferentes entradas.

---

## Salvamento e abertura de consultas SQL

As consultas podem ser salvas em seu PC, em um servidor de arquivos ou em um servidor de banco de dados.

### Salvamento de consultas SQL em arquivos

1. A partir de uma consulta aberta, clique no botão **Salvar** na barra de ferramentas.



Se a consulta já estiver salva, ela será salva novamente. Caso não tenha sido salva anteriormente, a caixa de diálogo Salvar Como é aberta.

2. Digite o nome do arquivo onde deseja armazenar a consulta e clique em **OK**. A consulta é salva.

### Aberturas de arquivos de consulta SQL salvos

1. Clique em **Abrir** na barra de ferramentas.



A caixa de diálogo Abrir é aberta.

2. Selecione o arquivo que deseja abrir e clique em **OK**. A consulta orientada é aberta em um novo documento de consulta.

## Salvamento de consultas SQL no servidor de banco de dados

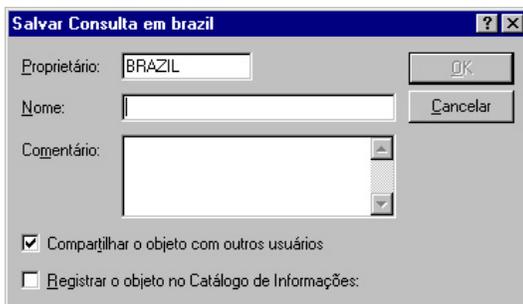
As consultas salvas no servidor podem ser acessadas por outros usuários. Caso deseje compartilhar suas consultas com outros usuários, salve-as no servidor de banco de dados.

- 1.

A partir de uma consulta aberta, clique no botão **Salvar no Servidor** na barra de ferramentas.



A caixa de diálogo Salvar Consulta é aberta.



2. Digite um proprietário, um nome, selecione se irá ou não compartilhar a consulta salva com outros usuários e clique em **OK**. A consulta é salva no servidor.

Caso já exista uma consulta com este nome, você será orientada a substituir a consulta previamente existente.

## Abertura de consultas SQL salvas no servidor de banco de dados

Para abrir consultas salvas no servidor de banco de dados.

- 1.

Clique no botão **Abrir a Partir do Servidor** na barra de ferramentas.



A caixa de diálogo Abrir a Partir do Servidor é aberta.



2. Digite um servidor, proprietário e nome e clique em **OK**. A consulta SQL é aberta.

---

## Impressão de consultas SQL

Você pode visualizar e imprimir suas consultas SQL.

### Visualização de uma consulta

1. Abra uma consulta e ative a exibição SQL. A instrução SQL aparece.
2. A partir do menu **Arquivo**, clique em **Configuração da Página**. A caixa de diálogo Configuração da Página é aberta.
3. Faça quaisquer alterações que deseje no layout da página e clique em **OK**.
4. Clique no botão **Visualizar Impressão** na barra de ferramentas.



Uma prévia da consulta impressa aparece.

### Impressão de consultas SQL

1. Abra uma consulta e ative a exibição SQL. A instrução SQL aparece.
2. A partir do menu **Arquivo**, clique em **Configuração da Página**. A caixa de diálogo Configuração da Página é aberta.
3. Faça quaisquer alterações que deseje no layout da página e clique em **OK**.
4. Clique no botão **Imprimir** na barra de ferramentas.



A consulta é impressa.



---

## Capítulo 3. Operação com Consultas Orientadas

As consultas orientadas são uma forma fácil de se criar uma consulta através da seleção de opções dos menus e listas. Após criar uma consulta orientada, você pode salvá-la ou convertê-la em consulta SQL.

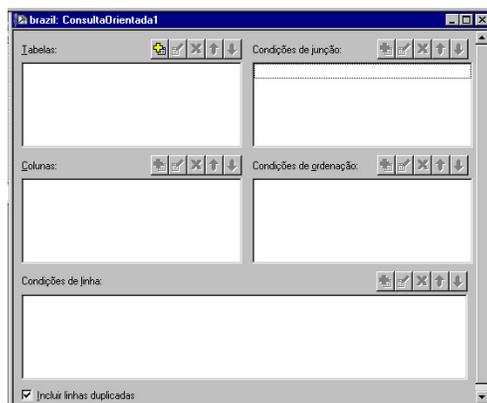
---

### Construção de Consultas Simples

Consultas simples podem ser construídas facilmente utilizando a interface de consulta orientada.

#### Abertura de novas consultas orientadas

- A partir do menu **Arquivo**, clique em **Nova Consulta Orientada**. Um novo documento de consulta orientada é aberto.



#### Botões de ação da consulta orientada

Edite as consultas orientadas utilizando os botões de ação de consulta. Um conjunto de botões aparece acima da seção que ele controla.

Botão de ação de consulta orientada	Aparência	Resultado
Incluir		Clique para incluir um item na consulta orientada.
Editar		Clique para editar o item destacado na consulta.
Eliminar		Clique para eliminar o item selecionado.

Mover para cima e  
Mover para baixo



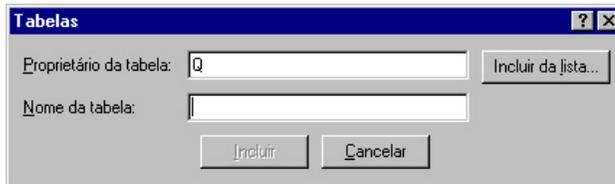
Clique para mover o item selecionado para cima e para baixo na consulta orientada.

## Inclusão de tabelas e consultas orientadas

1. Na seção Tabelas do documento de consulta orientada, clique no botão **Incluir**.



A caixa de diálogo Tabelas é aberta.



- 2.

Digite o proprietário e o nome da tabela que deseja incluir e clique em **Incluir**. A tabela é incluída na consulta.

**Nota:** Você pode usar padrões para selecionar objetos de uma lista de objetos correspondentes.

- Use o caractere de porcentagem (%) para representar cadeia de qualquer tamanho que contenha qualquer caractere. Por exemplo, para listar todas as tabelas com um nome começando com a letra A, digite A%.
- Use o caractere de sublinhado (\_) para representar um caractere individual. Por exemplo, para listar todas as tabelas com um proprietário que possui a letra A na segunda posição, você digita \_A%.

Depois de fornecer um padrão, clique em **Incluir da Lista** e selecione uma tabela da lista resultante.

3. Inclua quaisquer condições adicionais de tabelas na consulta e clique em **Fechar**. O documento consulta orientada aparece com as novas tabelas listadas.

## Execução de consultas orientadas

Você executa uma consulta orientada da mesma maneira que você executa uma consulta SQL. Clique no botão Executar Consulta na barra de ferramentas.



A consulta orientada é executada.

## Construção de Consultas Complexas

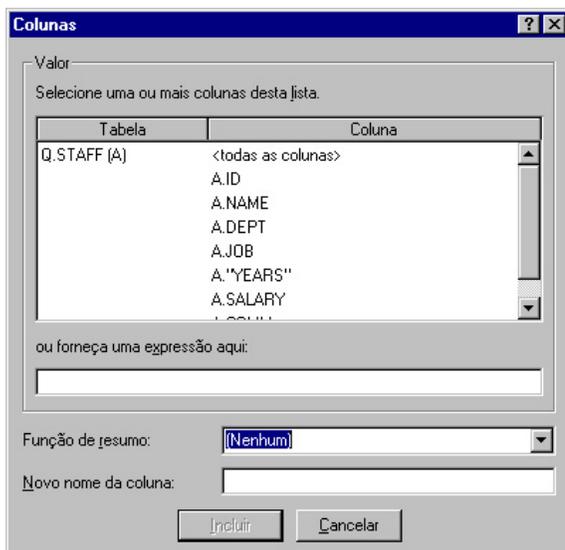
Consultas mais complexas também podem ser construídas utilizando a interface de consulta orientada.

### Inclusão de colunas em consultas orientadas

1. Na seção Colunas do documento Consulta Orientada, clique no botão **Incluir**.



A caixa de diálogo Colunas é aberta.



2. Selecione a coluna que você deseja incluir e clique em **Incluir**. A coluna é incluída na consulta orientada.
3. Inclua quaisquer colunas adicionais na consulta e clique em **Fechar**. O documento Consulta Orientada aparece com as novas colunas listadas.

**Nota:** Uma função de sumário pode ser aplicada à coluna, selecionando-se uma no campo Função. As funções de sumário disponíveis são: AVERAGE, COUNT, MAXIMUM, MINIMUM e SUM.

**Nota:** Para renomear uma coluna na consulta, digite um novo nome de coluna no campo **Novo nome da coluna**.

## Uso de condições de ordenação

As condições de ordenação são usadas para especificar o modo em que você ordena as linhas na consulta. As linhas podem ser classificadas na ordem ascendente (A-Z) ou descendente (Z-A).

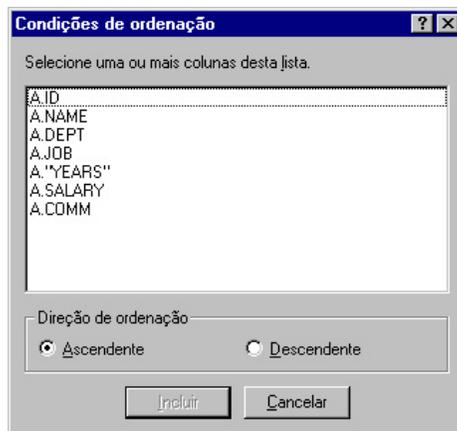
Se você classificar as suas linhas através de mais de uma coluna, a primeira coluna será classificada primeiro, a segunda coluna será classificada dentro da ordem da primeira coluna e assim por diante.

## Inclusão de condições de ordenação

1. Na seção Condições de Ordenação do documento Consulta Orientada, clique no botão **Incluir**.



A caixa de diálogo Condições de Ordenação aparece.



2. Selecione a coluna pela qual deseja classificar, a direção da ordenação e clique em **Incluir**. A condição de ordenação é incluída na consulta orientada.
3. Inclua quaisquer condições adicionais de ordenação na consulta e clique em **Fechar**. O documento Consulta Orientada aparece com as novas condições de ordenação listadas.

## Uso de condições de linha

Muitas vezes você deseja exibir somente certas linhas em uma tabela. Para especificar linhas a exibir, inclua condições de linha. Se você não usar condições de linha, todas as linhas na tabela serão exibidas.

As seguintes condições de linha estão disponíveis:

- Igual a

- Menor que
- Menor que ou igual a
- Maior que
- Maior que ou igual a
- Entre
- Inicializar com
- Terminar com
- Contendo
- NULO

As condições de linha são controladas pelos seguintes operadores:

- É
- Não é

### Inclusão de condições de linha

1. Na seção Condições de Linha do documento Consulta Orientada, clique no botão **Incluir**.



A caixa de diálogo Condições de Linha é aberta.

Coluna	Tipo de Dados
A.ID	NUMERIC
A.NAME	CHARACTER
A.DEPT	NUMERIC
A.JOB	CHARACTER
A."YEARS"	NUMERIC
A.SALARY	NUMERIC
A.COMM	NUMERIC

2. Selecione as partes da instrução condicional e clique em **Incluir**.

Parte da condição de linha	Função
Lado esquerdo	Selecionar a coluna que você deseja examinar.
Operador	Determinar a relação entre os lados esquerdo e direito da linha.

A condição de linha é incluída na consulta orientada.

3. Inclua quaisquer condições adicionais de linha na consulta e clique em **Fechar**. O documento Consulta Orientada aparece com as novas condições de linha listadas.

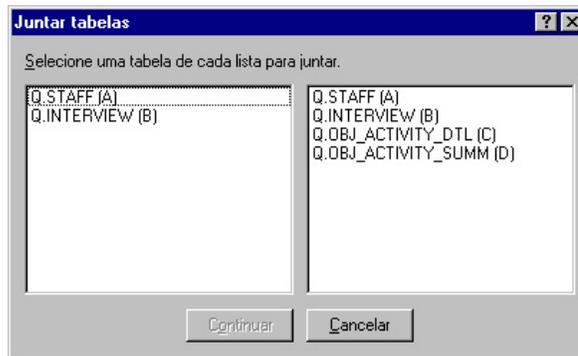
## Uso de múltiplas tabelas em consultas orientadas

Você pode incluir informações de mais de uma tabela em uma consulta orientada.

É necessário relacionar as duas tabelas, especificando uma ou mais condições de junção em cada uma delas. Somente as linhas das tabelas onde as colunas de junção são iguais são incluídas nos resultados. O tipo de dados de cada coluna em uma condição de junção deve corresponder. Uma vez que você especificou uma relação entre duas colunas, o QMF para Windows lembra-se da relação e, a sugere nas consultas futuras, tornando a criação de consultas subsequentes mais simples e mais eficiente.

## Criação de condições de junção de consulta orientada

1. Na seção Tabelas da Janela Consulta Orientada, clique no botão **Incluir** para incluir pelo menos duas tabelas. Se tabelas nunca foram unidas, a caixa de diálogo Juntar Tabelas é aberta. Caso tenham sido, o QMF para Windows sugere a condição de junção que foi usada anteriormente.



2. Selecione uma coluna com o mesmo tipo de dados a partir de cada tabela e clique em **Incluir**. A nova condição de junção aparece na consulta orientada.

---

## Consultas Orientadas e SQL

Você pode utilizar a interface de consulta orientada para aprender SQL.

## Exibição de SQL para consultas orientadas

A partir da exibição da consulta orientada, clique no botão **Exibir SQL** na barra de ferramentas.



A instrução SQL equivalente para a consulta orientada aparece. Você não pode modificar a instrução SQL nessa exibição.

## Conversão de consultas orientadas em SQL

Você pode converter uma consulta orientada em um novo documento de consulta SQL. A nova consulta SQL pode ser modificada, salva, impressa e executada. A partir do menu **Consulta**, clique em **Converter em SQL**. A consulta é convertida em um novo documento de consulta SQL.

---

## Uso de Variáveis de Substituição nas Consultas Orientadas

As variáveis de substituição podem ser usadas em uma consulta orientada da mesma maneira que em uma consulta SQL. Consulte “Variáveis de Substituição em Consultas SQL” .

Por exemplo, as variáveis de substituição podem ser usadas em:

- uma condição de linha  
DEPT É Maior Que Ou Igual A &MinDept
- uma especificação de coluna  
&InputNum

---

## Salvamento de Consultas Orientadas

As consultas orientadas podem ser salvas em arquivos no seu PC, em um servidor de arquivos ou em um servidor de banco de dados.

### Salvamento de consultas orientadas em arquivos

1. A partir de uma consulta orientada aberta, clique no botão **Salvar** na barra de ferramentas.



- Nota:** Se a consulta já estiver salva, ela será salva novamente. Caso não tenha sido salva anteriormente, a caixa de diálogo Salvar Como é aberta.
2. Digite o nome do arquivo onde deseja que a consulta orientada seja armazenada e clique em **OK**. A consulta é salva.

## Abertura de arquivos salvos de consulta orientada

1. Clique em **Abrir** na barra de ferramentas.



A caixa de diálogo Abrir é aberta.

2. Selecione o arquivo que deseja abrir e clique em **OK**. A consulta orientada selecionada é aberta em um novo documento de consulta.

## Salvamento de consultas orientadas no servidor de banco de dados

1. A partir de uma consulta orientada aberta, clique no botão **Salvar no Servidor** na barra de ferramentas.



A caixa de diálogo Salvar Consulta é aberta.

2. Digite um proprietário, um nome, selecione se irá ou não compartilhar a consulta salva com outros usuários e clique em **OK**. A consulta é salva no servidor.

Se já existir uma consulta com este nome, você será avisado sobre se deseja substituir a consulta existente.

## Abertura de consultas orientadas salvas no servidor de banco de dados

Para abrir consultas orientadas salvas no servidor de banco de dados.

- 1.

Clique no botão **Abrir a Partir do Servidor** na barra de ferramentas.



A caixa de diálogo Abrir a Partir do Servidor é aberta.



2. Digite um servidor, proprietário e nome e clique em **OK**. A consulta orientada é aberta.

---

## Impressão de Consultas Orientadas

Você pode imprimir a consulta orientada. Você também pode imprimir o texto SQL de uma consulta orientada. Consulte “Impressão de consultas SQL” na página 15.

### Visualização de consultas orientadas

Os resultados ou o texto de uma consulta orientada podem ser visualizados antes de serem impressos.

1. Abra uma consulta e ative a exibição orientada. A consulta aparece.
2. A partir do menu **Arquivo**, clique em **Configuração da Página**. A caixa de diálogo Configuração da Página é aberta.
3. Faça as alterações que deseja no layout da página e clique em **OK**.
4. Clique no botão **Visualizar Impressão** na barra de ferramentas.



Uma visualização da consulta impressa aparece.



---

## Capítulo 4. Operação com Resultados de Consultas

Você pode aplicar formatação, agrupamento e agregação diretamente a resultados de consultas. Esta formatação pode ser salva com a consulta ou exportada como um formulário.

---

### Classificação e dimensionamento de resultados de consultas

Os usuários podem selecionar, redimensionar, reordenar e classificar os resultados de dados de uma consulta.

#### Seleção de colunas e linhas

Depois que uma consulta é executada, você pode utilizar os controles na exibição Resultados para editar e selecionar informações.

Seletores de colunas e de linhas	Aparência	Função
Seletor de linhas		Clique para selecionar todos os dados em uma linha.
Seletor de colunas		Clique para selecionar todos os dados em uma coluna.
Célula		Clique diretamente em uma célula para selecioná-la.
Botões Mover para baixo e Mover para cima		Clique a fim de deslocar para a parte superior ou inferior de um conjunto de resultados de consulta.

#### Redimensionamento de colunas e linhas

Você pode alterar a aparência de um conjunto de resultados de consulta, redimensionando suas colunas e linhas.

1. Utilizando o mouse, selecione a linha divisória preta entre duas colunas ou linhas.
2. Arraste a linha divisória de um lado para outro ou para cima e para baixo para redimensionar a coluna ou linha.

**Nota:** Caso você salve uma consulta depois de redimensionar suas linhas ou colunas, esta consulta sempre será exibida utilizando a nova formatação.

## Ajuste automático de colunas e linhas

Você pode dimensionar automaticamente colunas e linhas para encaixar os dados que elas contém.

Utilizando o mouse, selecione uma coluna ou linha inteira e dê um clique duplo na linha divisória preta entre ela e o objeto adjacente. A coluna ou linha é redimensionada automaticamente para ajustar-se a seus dados.

**Nota:** Caso você salve uma consulta depois de redimensionar suas linhas ou colunas, esta consulta sempre será exibida utilizando a nova formatação.

## Classificação de resultados da consulta

Depois que uma consulta é executada, você pode classificar os resultados em ordem alfabética por coluna.

Na exibição Resultados de uma consulta, selecione uma coluna e selecione **Classificação Crescente** no menu **Resultados**.

Os resultados da consulta são classificados em ordem crescente.

-ou-

Na exibição Resultados de uma consulta, selecione uma coluna e selecione **Classificação Decrescente** no menu **Resultados**.

Os resultados da consulta são classificados em ordem decrescente.

**Nota:** Para aplicar uma classificação mais complexa à coluna selecionada, escolha **Classificar** no menu **Resultados**.

## Reordenação de colunas

Você pode alterar a ordem das colunas no resultado da consulta.

Na exibição Resultados de uma consulta, selecione uma coluna e arraste-a para sua nova localização.

A coluna aparece na nova ordem.

---

## Formatação de resultados da consulta

A fonte utilizada para exibição de consultas e de resultados de consulta. A opção de fontes varia de acordo com o que está instalado em seu computador. Para obter mais informações sobre a inclusão de fontes, consulte o recurso de ajuda do sistema operacional.

**Nota:** Caso você salve uma consulta após selecionar uma nova fonte para exibição de resultados da consulta, estes resultados sempre serão exibidos utilizando a nova fonte.

### **Seleção da fonte de exibição dos resultados da consulta**

1. Na exibição Resultados, selecione **Definir Fonte** no menu **Resultados**. A caixa de diálogo Fonte é aberta.
2. Selecione o tipo e o tamanho da fonte para exibição dos resultados da consulta e clique em **OK**. Os resultados da consulta serão exibidos no formato especificado.

**Nota:** Clique em **Definir Como Padrão** para utilizar a fonte selecionada como fonte padrão para todos os resultados da consulta.

### **Formatação de resultados de consulta numéricos**

1. Na exibição Resultados de uma consulta, selecione uma coluna que contenha valores numéricos e selecione **Formatar** no menu **Resultados**. A caixa de diálogo Formatar é aberta.
2. Especifique a formatação que deseja aplicar e clique em **OK**. Os valores são formatados de acordo com a sua seleção.

**Nota:** Clique em **Definir Como Padrão** para utilizar a fonte selecionada como fonte padrão para todos os resultados da consulta.

### **Conversão da formatação de resultados da consulta em um formulário**

Você pode converter a formatação dos resultados da consulta em um formulário.

1. No menu **Resultados**, selecione **Exibir Relatório**.  
A caixa de diálogo Selecionar Formulário é aberta.
2. Selecione **Da Consulta** e clique em **OK**.  
A formatação dos resultados da consulta é convertida para um formulário e aberta em uma nova janela de formulário.

---

### **Agrupamento e agregação de resultados da consulta**

Você pode aplicar formatação de agrupamento, agregação e resumo aos resultados da consulta.

#### **Agrupamento de resultados da consulta**

Você pode agrupar resultados da consulta com ou sem informações de resumo.

1. Selecione a coluna que deseja agrupar.
2. No menu **Resultados**, selecione o tipo de agrupamento que deseja aplicar.  
A coluna é agrupada de acordo com a sua seleção.

## Resumo de resultados da consulta

Você pode resumir os resultados da consulta por coluna.

1. Selecione a coluna que deseja agrupar.
2. No menu **Resultados**, selecione o tipo de resumo que deseja aplicar.  
A coluna é resumida de acordo com a sua seleção.

---

## Salvamento de resultados da consulta e formatação

Você pode salvar os resultados da consulta e salvar a formatação como um formulário.

### Salvamento dos resultados da consulta como uma tabela

Você pode salvar os resultados da consulta como uma tabela em um servidor de banco de dados.

1. No menu **Resultados**, selecione **Salvar em Banco de Dados**.  
A caixa de diálogo Salvar Dados é aberta.
2. Digite um nome de proprietário e de tabela e clique em **OK**.  
Os resultados da consulta são salvos como uma tabela no banco de dados.

### Salvamento de resultados da consulta em arquivos

Você pode salvar os resultados da consulta em um arquivo em seu PC ou em um servidor de arquivos.

1. No menu **Resultados**, selecione **Salvar em Arquivo**.  
A caixa de diálogo Exportar Dados é aberta.
2. Especifique a localização onde deseja salvar o arquivo, quaisquer opções de exportação e clique em **OK**.  
Os resultados da consulta são salvos em um arquivo.

---

## Impressão de resultados da consulta

Você pode visualizar e imprimir os resultados de suas consultas.

### Visualização de resultados da consulta

1. Abra uma consulta e execute-a. Os resultados da consulta aparecem.
2. No menu **Arquivo**, selecione **Configuração da Página**. A caixa de diálogo Configuração da Página é aberta.
3. Faça quaisquer alterações que deseje no layout da página e clique em **OK**.
4. Clique no botão **Visualizar Impressão** na barra de ferramentas.



Uma prévia dos resultados de consulta impressos aparece.

## Impressão de resultados da consulta

1. Abra uma consulta e ative a exibição Resultados. Os resultados da consulta aparecem.
2. No menu **Arquivo**, selecione **Configuração da Página**. A caixa de diálogo Configuração da Página é aberta.
3. Faça quaisquer alterações que deseje no layout da página e clique em **OK**.
4. Clique no botão **Imprimir** na barra de ferramentas.



Os resultados da consulta são impressos.



---

## Capítulo 5. Operação com Relatórios

Os relatórios são criados através da combinação de resultados de consulta com a formatação de um formulário.

---

### Formulários

Formulários são conjuntos de instruções de formatação utilizados para criar, exibir e imprimir relatórios.

#### Compreensão de Formulários

Os formulários são compostos de um número de componentes. Todos estes componentes podem ser editados em um documento de formulário.

##### Principal

Os componentes principais de um formulário, incluindo cabeçalhos, rodapés e quebras.

##### Quebras

Características, conteúdo e colocação de até seis linhas de subtotal num relatório.

##### Cálculos

Expressões de cálculo em relatório.

**Nota:** Para utilizar os cálculos de formulários, é necessário que o IBM ObjectREXX esteja instalado em sua máquina.

##### Colunas

Aparência e formatação de colunas no relatório. Entre as características definíveis estão ordem da coluna, formato, utilização, recuo e largura.

##### Condições

Restrições condicionais de formatação. Pode-se, por exemplo, definir o formulário para que não sejam exibidas linhas que não satisfazem a determinadas características.

##### Detalhes

Títulos de detalhes e texto do corpo do relatório. É aí que podem ser associados e substituídos dados tabulares por texto de formato livre para criar cartas de formulário ou etiquetas de endereço.

**Final** Conteúdo e colocação do texto final do relatório. É possível, por exemplo, optar por incluir o texto final e os dados de resumo no final do relatório.

## HTML

Conteúdo e colocação de tags HTML e formatação em relatórios HTML.

## Opções

Diversas opções de aparência para seu relatório.

## Página

Conteúdo e colocação de títulos e rodapés de página no relatório.

## Produção de um relatório usando um formulário

Os relatórios são criados através da combinação de resultados de consulta com as opções de formatação contidas em um formulário. Vários relatórios podem ser produzidos a partir de um único conjunto de resultados de consulta, repetindo-se este processo.

1. A partir de uma exibição de resultados de consulta, clique no botão **Exibir Relatório**.



A caixa de diálogo Selecionar Formulário é aberta.



2. Dependendo do tipo de formulário selecionado na caixa de diálogo Selecionar a Partir De, você será solicitado a fornecer informações adicionais. Especifique a localização do arquivo ou proprietário e nome ou título do documento, conforme apropriado e clique em **OK**. O relatório é gerado utilizando o formulário selecionado e os atuais resultados da consulta.

## Edição de um formulário

A janela Formulário apresenta várias opções para edição e formatação de formulários.

A partir de um formulário aberto, exiba o menu Formulário. O menu Formulário lista todas as suas opções para edição e formatação do seu formulário. Qualquer um destes componentes também pode ser editado, clicando-se no botão correspondente da barra de ferramentas.

---

## Criação de Formulário

Todas estas etapas incluem dados de exemplo da tabela Q.STAFF. Tente experimentar com definições diferentes para criar os seus próprios relatórios personalizados.

### Passo 1: Criar um formulário

1. Execute a seguinte consulta SQL para recuperar os dados que serão exibidos no relatório:

```
SELECT * FROM Q.STAFF ORDER BY DEPT, NAME
```

Os resultados da consulta aparecem.

2. Clique no botão **Exibir Relatório** na barra de ferramentas. A caixa de diálogo Seleccionar Formulário é aberta.
3. Especifique que deseja utilizar o formulário padrão e clique em **OK**. O QMF para Windows exibe o relatório padrão. Para fazer alterações no formato padrão, clique nos botões do componente de formulário na barra de ferramentas. Um botão para cada componente de formulário é exibido na barra de ferramentas do formulário.

### Passo 2: Alterar a ordem das colunas

Queremos que NAME seja a primeira coluna do relatório e que ID seja a segunda. A ordem das colunas é especificada no componente Colunas do formulário.

1. Clique em **Colunas...** no menu **Formulário** para exibir a guia Colunas da caixa de diálogo Formulário.
2. Altere a seqüência de uma coluna digitando sobre o valor de seqüência existente. Para tornar NAME a primeira coluna do relatório, mude seu número de seqüência (coluna chamada Seq na lista) para 1.
3. Para tornar ID a segunda coluna do relatório, mude seu número de seqüência para 2 e clique em **OK**. O QMF para Windows exibe o relatório com a nova ordem das colunas na janela Formulário.

### Passo 3: Alterar os títulos das colunas

O primeiro título da coluna deve ser EMPLOYEE e o segundo COMMISSION. O texto do cabeçalho da coluna é especificado no componente Colunas do formulário.

1. Clique em **Colunas...** no menu **Formulário** para exibir a guia Colunas da caixa de diálogo Formulários.
2. Mude o título da coluna digitando sobre o texto do título já existente. Altere o primeiro título da coluna para EMPLOYEE e o último para COMMISSION e clique em **OK**. O QMF para Windows exibe o relatório com os novos títulos de coluna na janela Formulário.

#### Passo 4: Alterar o formato da coluna

Queremos que a coluna SALARY seja exibida com o símbolo monetário adequado. O formato de uma coluna é determinado pelo código de edição, que é especificado no componente Colunas do formulário.

1. Clique em **Colunas...** no menu **Formulário** para exibir a guia Colunas da caixa de diálogo Formulários.
2. Altere o código de edição da coluna SALARY para D2, digitando sobre o código já existente e clique em **OK**. O QMF para Windows exibe o relatório com a coluna SALARY usando o símbolo monetário adequado na janela Formulário.

#### Passo 5: Incluir informações de resumo

Nós queremos dividir o relatório em seções separadas, uma seção para cada departamento. E queremos ver o total de SALARY e COMMISSION para cada departamento no final de cada seção. Para isso, precisamos especificar como as colunas do relatório devem ser usadas. O uso de uma coluna é determinado pelo código de uso, e é especificado no componente Colunas do formulário.

1. Clique em **Colunas...** no menu **Formulário** para exibir a guia Colunas da caixa de diálogo Formulários.
2. Para dividir o relatório em seções baseadas no DEPT, altere o código de uso do DEPT para BREAK1. Os códigos de utilização que têm a palavra BREAK produzem uma quebra de seção para a coluna especificada. O número que vem depois da palavra BREAK determina o nível da quebra; cada relatório suporta até seis níveis de quebra.
3. Para especificar que desejamos incluir no total de SALARY e COMMISSION para cada DEPT, altere o código de uso de SALARY e COMMISSION para SUM.
4. O relatório ficará mais fácil de ser compreendido se também incluirmos informações descritivas no final de cada quebra de seção. Para fazer isso, clique em **Quebras...** no menu **Formulário**.
5. Especifique texto de rodapé com quebra na guia Quebras da caixa de diálogo Formulário. Defina a primeira linha do rodapé da quebra para Total do Departamento e clique em **OK**. O QMF para Windows exibe a janela Formulário.

#### Passo 6: Incluir cabeçalhos e rodapés de páginas

Queremos colocar um título e um rodapé de página no relatório. Os cabeçalhos e rodapés de página são especificados no componente Página do formulário.

1. Clique em **Página...** no menu **Formulário** para exibir a guia Página da caixa de diálogo Formulário.

2. A parte superior desse diálogo é usada para especificar o título da página. Defina a primeira linha do título da página para Relatório do Departamento e a segunda para Salário e Comissões Totais. Escolha como o título deve ser alinhado.
3. A parte de baixo desse diálogo é usada para especificar o rodapé da página. Defina a primeira linha do rodapé da página para Fim da Página. Escolha como o rodapé deve ser alinhado e clique em **OK**. O QMF para Windows exibe a janela Formulário.

---

## Salvamento de formulários

Os formulários podem ser salvos no seu PC, em um servidor de arquivos ou em um servidor de banco de dados.

### Salvamento de um formulário em um arquivo

1. A partir de um formulário aberto, clique no botão **Salvar**.
2. Se o formulário foi salvo anteriormente, selecione **Salvar**. Caso não tenha sido salvo anteriormente, a caixa de diálogo Salvar Como é aberta.
3. Digite o nome do arquivo onde deseja que o formulário seja armazenado e clique em **OK**. O formulário é salvo.

### Abertura de arquivos de formulário salvos

1. Clique em **Abrir** na barra de ferramentas.



A caixa de diálogo Abrir é aberta.

2. Selecione o arquivo que deseja abrir e clique em **OK**. O formulário selecionado é aberto em um novo documento de formulário.

### Salvamento de formulários no servidor de banco de dados

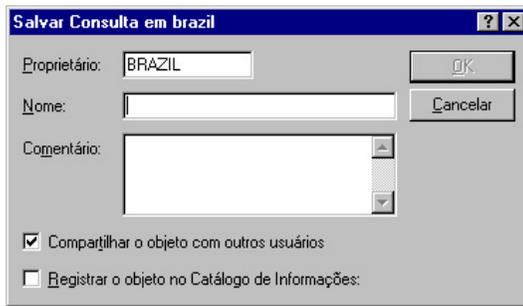
Os formulários salvos no servidor podem ser acessados por outros usuários. Caso deseje compartilhá-los com outros usuários, salve-os no servidor de banco de dados.

- 1.

A partir de um formulário aberto, clique no botão **Salvar no Servidor** na barra de ferramentas.



A caixa de diálogo Salvar Formulário é aberta.



2. Digite um proprietário, um nome, selecione se irá ou não compartilhar o formulário salvo com outros usuários e clique em **OK**. O formulário é salvo no servidor.

Se já existir um formulário com este nome, você será consultado sobre se deseja substituir o formulário existente.

### Abertura de formulários salvos no servidor de banco de dados

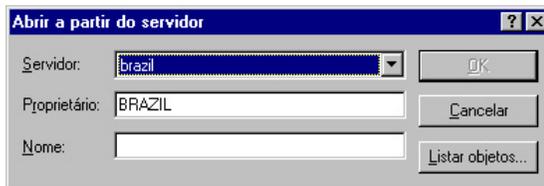
Para abrir formulários salvos no servidor de banco de dados.

- 1.

Clique no botão **Abrir a Partir do Servidor** na barra de ferramentas.



A caixa de diálogo Abrir a Partir do Servidor é aberta.



2. Digite um servidor, proprietário e nome e clique em **OK**. O formulário é aberto.

---

## Impressão de relatórios

Você pode produzir relatórios impressos.

1. Abra um formulário e clique em **Configuração da Página**.
2. Faça quaisquer alterações que deseje no layout da página e clique em **OK**.
3. Clique em **Imprimir Relatório** no menu **Arquivo**.  
O relatório é impresso.

---

## Exportação de relatórios

Um relatório pode ser exportado para um arquivo.

1. Abra um formulário e clique em **Configuração da Página**.
2. Faça quaisquer alterações que deseje no layout da página e clique em **OK**.
3. Clique em **Exportar Relatório** no menu **Arquivo**. A caixa de diálogo Exportar Relatório é aberta.



4. Digite o nome do arquivo onde deseja armazenar o relatório e clique em **OK**. O relatório é exportado.



---

## Capítulo 6. Operação com Procedimentos

Os procedimentos lineares possibilitam a execução de consultas, geração de relatórios, edição de dados e realização de outras funções através da execução de um único comando. Para obter uma listagem completa de todos os comandos de procedimento suportados pelo QMF para Windows, consulte o recurso de ajuda online.

Os procedimentos com lógica, ou procedimentos REXX, são semelhantes a procedimentos lineares, mas contêm a linguagem de programação Object REXX da IBM, bem como comandos de procedimentos. O Object REXX deve estar instalado localmente para se executar procedimentos com lógica.

---

### Execução de Procedimentos

Os procedimentos são utilizados para executar múltiplas funções com um comando.

#### Criação de um novo procedimento linear

A partir do menu **Arquivo**, selecione **Novo Procedimento**.

Um novo documento de procedimento é aberto.

#### Criação de um novo procedimento com lógica

1. A partir do menu **Arquivo**, selecione **Novo Procedimento**.  
Um novo documento de procedimento é aberto.
2. Digite uma linha de comentário REXX na primeira linha do procedimento. As linhas de comentário REXX começam com `/*` e terminam com `*/`
3. Digite quaisquer comandos de procedimentos do QMF que desejar no procedimento. Os comandos do QMF devem ser digitados em maiúsculas e devem estar entre aspas.
4. Digite quaisquer comandos REXX que desejar no procedimento.

**Nota:** Os comandos REXX são executados localmente, não no servidor do banco de dados. O Object REXX deve estar instalado localmente.

#### Execução de um procedimento em um servidor de banco de dados

1. Abra um novo documento de procedimento e digite uma série de comandos ou abra um procedimento existente a partir de um arquivo ou servidor de banco de dados.

2. Clique no botão **Executar Procedimento** na barra de ferramentas.



O procedimento é executado.

---

## Salvamento de Procedimentos

Os procedimentos podem ser salvos no seu PC, em um servidor de arquivos ou em um servidor de banco de dados.

### Salvamento de procedimentos em arquivos

1. A partir de um procedimento aberto, clique no botão **Salvar** na barra de ferramentas.



Caso o procedimento tenha sido salvo antes, o procedimento é salvo. Caso não tenha sido salvo anteriormente, a caixa de diálogo **Salvar Como** é aberta.

2. Digite o nome do arquivo onde deseja que o procedimento seja armazenado e clique em **OK**. O procedimento é salvo.

### Abertura de um arquivo de procedimento salvo

1. Clique em **Abrir** na barra de ferramentas.



A caixa de diálogo **Abrir** aparece.

2. Selecione o arquivo que deseja abrir e clique em **OK**. O procedimento selecionado é aberto em um novo documento de procedimento.

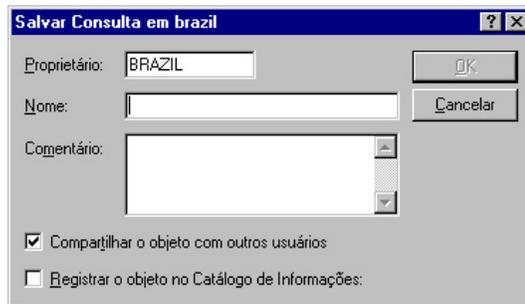
### Salvamento de um procedimento no servidor de banco de dados

- 1.

A partir de um procedimento aberto, clique no botão **Salvar no Servidor** na barra de ferramentas.



A caixa de diálogo Salvar Procedimento é aberta.



2. Digite um proprietário, um nome, selecione se irá ou não compartilhar o procedimento salvo com outros usuários e clique em **OK**. O procedimento é salvo no servidor.

Caso já exista um procedimento com este nome, você será orientado a substituir o procedimento existente.

### Abertura de procedimentos salvos no servidor de banco de dados

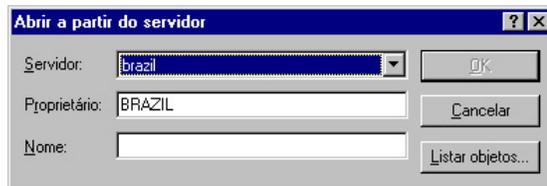
Para abrir procedimentos salvos no servidor de banco de dados.

- 1.

Clique no botão **Abrir a Partir do Servidor** na barra de ferramentas.



A caixa de diálogo Abrir a Partir do Servidor é aberta.



2. Forneça um servidor, proprietário e nome e clique em **OK**. O procedimento é aberto.

---

## Impressão de Procedimentos

O texto de um procedimento pode ser impresso.

### Visualização de um procedimento

1. Abra um procedimento. Os comandos do procedimento aparecem.
2. A partir do menu **Arquivo**, clique em **Configuração da Página**. A caixa de diálogo Configuração da Página é aberta.

3. Faça quaisquer alterações que deseje no layout da página e clique em **OK**.
4. Clique no botão **Visualizar Impressão** na barra de ferramentas:



Uma prévia do procedimento impresso aparece.

### **Impressão de um procedimento**

1. Abra um procedimento. Os comandos do procedimento aparecem.
2. A partir do menu **Arquivo**, clique em **Configuração da Página**. A caixa de diálogo Configuração da Página é aberta.
3. Faça quaisquer alterações que deseje no layout da página e clique em **OK**.
4. Clique no botão **Imprimir** na barra de ferramentas:



O procedimento é impresso.

---

## Capítulo 7. Operação com Listas

As listas oferecem uma forma fácil de visualização de coleções de objetos do QMF.

---

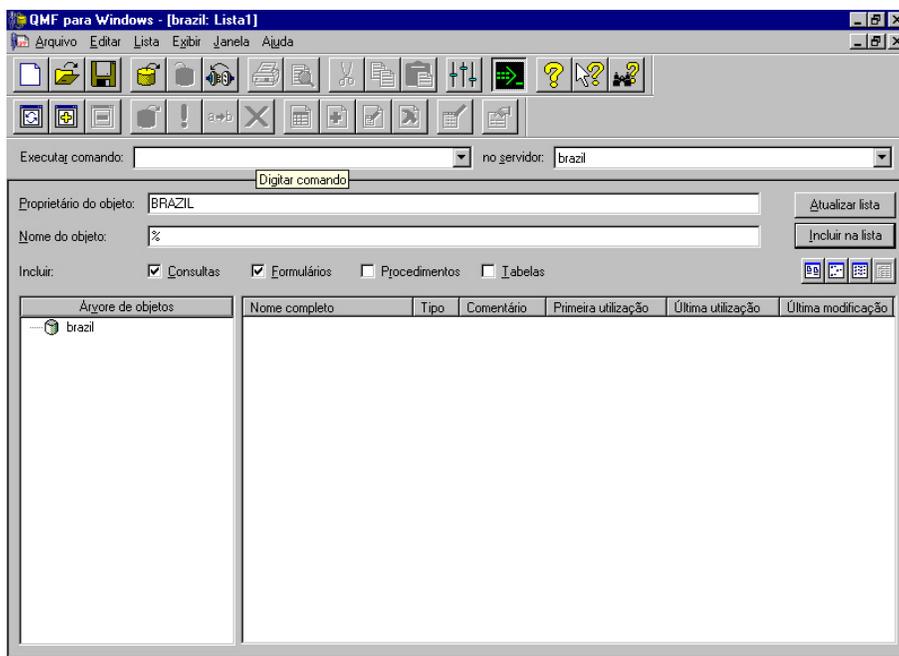
### Objetos

O QMF para Windows reconhece quatro tipos de objetos: consultas, formulários, procedimentos e tabelas. A janela Lista pode ser utilizada para visualizar os objetos baseados no nome, proprietário e tipo do objeto.

#### Lista de objetos

1. No menu **Arquivo**, selecione **Nova Lista**.

A janela Lista é aberta.



2. Especifique um proprietário e um nome.

**Nota:** Você pode usar padrões para selecionar objetos de uma lista de objetos correspondentes.

- Use o caractere de porcentagem (%) para representar cadeia de qualquer tamanho que contenha qualquer caractere. Por exemplo, para listar todas as tabelas com um nome começando com a letra A, digite A%.

- Use o caractere de sublinhado (\_) para representar um caractere individual. Por exemplo, para listar todas as tabelas com um proprietário que possui a letra A na segunda posição, você digita \_A%.
3. Selecione o tipo de objeto que você está procurando.
  4. Clique em **Atualizar Lista**. Uma lista de objetos correspondentes salvos no servidor do banco de dados é exibida.

---

## Comandos da Janela Lista

Um clique com o botão direito do mouse sobre um objeto na janela Lista exibe uma lista de comandos idênticos àqueles no menu Lista.

### Exibição de objetos

Abra o objeto selecionado para visualização. Disponível para consultas, formulários, procedimentos e tabelas.

### Execução de objeto

Executa o objeto selecionado. Disponível para consultas e procedimentos.

### Desenhar objeto

Cria uma consulta com base na tabela selecionada. Você pode optar por desenhar uma consulta SQL SELECT, uma consulta SQL UPDATE, uma consulta SQL INSERT ou uma consulta orientada. Disponível para tabelas.

### Edição de objeto

Abre o objeto selecionado para edição. Disponível para tabelas.

### Propriedades

Exibe as propriedades do objeto selecionado, incluindo comentários, atributos, informações de uso do histórico. Disponível para consultas, formulários, procedimentos e tabelas.

---

## Criação de Listas

Listas podem ser criadas para servir como coleções de objetos. Por exemplo, você poderia criar uma lista de todas as consultas, formulários, procedimentos e tabelas relacionadas ao inventário para manter seu trabalho em um só lugar. Depois de criada, é possível incluir e remover objetos da lista e salvá-la para uso futuro.

### Inclusão de objetos em listas

Você pode incluir objetos em listas.

Em uma lista aberta, especifique as informações de proprietário e nome dos objetos que deseja incluir e clique no botão **Incluir na Lista**, na barra de

ferramentas.



Os objetos correspondentes ao proprietário e nome são incluídos na lista.

### Remoção de objetos das listas

Objetos não relacionados podem ser removidos das listas.

A partir de uma lista aberta, clique no botão **Remover** na barra de ferramentas.



O objeto é removido da lista, porém não é eliminado.

### Salvamento de listas em arquivos

1. A partir de uma lista aberta, clique no botão **Salvar** na barra de ferramentas.



Caso a lista tenha sido salva antes, a lista é salva. Caso não tenha sido salva anteriormente, a caixa de diálogo Salvar Como é aberta.

2. Digite o nome do arquivo onde deseja que a lista seja armazenada e clique em **OK**. A lista é salva.

### Abertura de arquivos de lista salvos

1. Clique em **Abrir** na barra de ferramentas.



A caixa de diálogo Abrir é aberta.

2. Selecione o arquivo que deseja abrir e clique em **OK**. A lista selecionada é aberta em um documento de lista.



---

## Capítulo 8. Operação com Arquivos de Job

Você pode programar e executar procedimentos utilizando arquivos de job. Os arquivos de job utilizam o programador do Windows para executar procedimentos de acordo com a data e hora predefinidas por você.

---

### Arquivos de job

Você pode criar arquivos de job e armazená-los localmente ou no servidor do banco de dados.

#### Criação de arquivos de job

1. No menu **Arquivo**, selecione **Novo Job**.

Um novo documento de job é aberto.

#### Execução de arquivos de job

Você pode executar arquivos de job que foram salvos localmente.

1. Abra um arquivo de job.
2. Clique no botão **Executar Job** na barra de ferramentas.



3. Arraste a linha divisória de um lado para outro ou para cima e para baixo para redimensionar a coluna ou linha.

**Nota:** Caso você salve uma consulta depois de redimensionar suas linhas ou colunas, esta consulta sempre será exibida utilizando a nova formatação.

#### Ajuste automático de colunas e linhas

Você pode dimensionar automaticamente colunas e linhas para encaixar os dados que elas contém.

Utilizando o mouse, selecione uma coluna ou linha inteira e dê um clique duplo na linha divisória preta entre ela e o objeto adjacente. A coluna ou linha é redimensionada automaticamente para ajustar-se a seus dados.

**Nota:** Caso você salve uma consulta depois de redimensionar suas linhas ou colunas, esta consulta sempre será exibida utilizando a nova formatação.

## Classificação de resultados da consulta

Depois que uma consulta é executada, você pode classificar os resultados em ordem alfabética por coluna.

Na exibição Resultados de uma consulta, selecione uma coluna e selecione **Classificação Crescente** no menu **Resultados**.

Os resultados da consulta são classificados em ordem crescente.

-ou-

Na exibição Resultados de uma consulta, selecione uma coluna e selecione **Classificação Decrescente** no menu **Resultados**.

Os resultados da consulta são classificados em ordem decrescente.

**Nota:** Para aplicar uma classificação mais complexa à coluna selecionada, selecione **Classificar** no menu **Resultados**.

## Reordenação de colunas

Você pode alterar a ordem das colunas no resultado da consulta.

Na exibição Resultados de uma consulta, selecione uma coluna e arraste-a para sua nova localização.

A coluna aparece na nova posição.

---

## Formatação de resultados da consulta

A fonte utilizada para exibição de consultas e de resultados de consulta. A opção de fontes varia de acordo com o que está instalado em seu computador. Para obter mais informações sobre a inclusão de fontes, consulte o recurso de ajuda do sistema operacional.

**Nota:** Caso você salve uma consulta após selecionar uma nova fonte para exibição de resultados da consulta, estes resultados sempre serão exibidos utilizando a nova fonte.

## Seleção da fonte de exibição dos resultados da consulta

1. Na exibição Resultados, selecione **Definir Fonte** no menu **Resultados**. A caixa de diálogo Fonte é aberta.
2. Selecione o tipo e o tamanho da fonte para exibição dos resultados da consulta e clique em **OK**. Os resultados da consulta serão exibidos no formato especificado.

**Nota:** Clique em **Definir Como Padrão** para utilizar a fonte selecionada como fonte padrão para todos os resultados da consulta.

### **Formatação de resultados de consulta numéricos**

1. Na exibição Resultados de uma consulta, selecione uma coluna que contenha valores numéricos e selecione **Formatar** no menu **Resultados**. A caixa de diálogo Formatar é aberta.
2. Especifique a formatação que deseja aplicar e clique em **OK**. Os valores

**Nota:** Clique em **Definir Como Padrão** para utilizar a fonte selecionada como fonte padrão para todos os resultados da consulta.

### **Conversão da formatação de resultados da consulta em um formulário**

Você pode converter a formatação dos resultados da consulta em um formulário.

1. No menu **Resultados**, selecione **Exibir Relatório**.  
A caixa de diálogo Selecionar Formulário é aberta.
2. Selecione Da consulta e clique em **OK**.  
A formatação dos resultados da consulta é convertida em um formulário e aberta em uma nova janela do formulário.

---

## **Agrupamento e agregação de resultados da consulta**

Você pode aplicar formatação de agrupamento, agregação e resumo aos resultados da consulta.

### **Agrupamento de resultados da consulta**

Você pode agrupar resultados da consulta com ou sem informações de resumo.

1. Selecione a coluna que deseja agrupar.
2. No menu **Resultados**, selecione o tipo de agrupamento que deseja aplicar.  
A coluna é agrupada de acordo com a sua seleção.

### **Resumo de resultados da consulta**

Você pode resumir os resultados da consulta por coluna.

1. Selecione a coluna que deseja agrupar.
2. No menu **Resultados**, selecione o tipo de resumo que deseja aplicar.  
A coluna é resumida de acordo com a sua seleção.

---

## **Salvamento de resultados da consulta e formatação**

Você pode salvar os resultados da consulta e salvar a formatação como um formulário.

## Salvamento dos resultados da consulta como uma tabela

Você pode salvar os resultados da consulta como uma tabela em um servidor de banco de dados.

1. No menu **Resultados**, selecione **Salvar em Banco de Dados**.

A caixa de diálogo Salvar Dados é aberta.

2. Digite um nome de proprietário e de tabela e clique em **OK**.

Os resultados da consulta são salvos como uma tabela no banco de dados.

## Salvamento de resultados da consulta em arquivos

Você pode salvar os resultados da consulta em um arquivo em seu PC ou em um servidor de arquivos.

1. No menu **Resultados**, selecione **Salvar em Arquivo**.

A caixa de diálogo Exportar Dados é aberta.

2. Especifique a localização onde deseja salvar o arquivo, quaisquer opções de exportação e clique em **OK**.

Os resultados da consulta são salvos em um arquivo.

---

## Impressão de resultados da consulta

Você pode visualizar e imprimir os resultados de suas consultas.

### Visualização de resultados da consulta

1. Abra uma consulta e execute-a. Os resultados da consulta aparecem.
2. No menu **Arquivo**, selecione **Configuração da Página**. A caixa de diálogo Configuração da Página é aberta.
3. Faça quaisquer alterações que deseje no layout da página e clique em **OK**.
4. Clique no botão **Visualizar Impressão** na barra de ferramentas:



Uma prévia dos resultados de consulta impressos aparece.

### Impressão de resultados da consulta

1. Abra uma consulta e ative a exibição Resultados. Os resultados da consulta aparecem.
2. No menu **Arquivo**, selecione **Configuração da Página**. A caixa de diálogo Configuração da Página aparece.
3. Faça quaisquer alterações que deseje no layout da página e clique em **OK**.

4. Clique no botão **Imprimir** na barra de ferramentas.



Os resultados da consulta são impressos.



---

## Capítulo 9. Operação com Consultas Estáticas

Uma consulta estática é uma consulta SQL que foi anteriormente passada para um servidor de banco de dados e ligada a um pacote. Quando uma consulta estática é executada, o servidor de banco de dados usa o texto SQL ligado ao pacote, e não o texto SQL que está aparecendo na janela de consulta. As consultas estáticas são mais eficientes quanto aos recursos do que as consultas dinâmicas, mas as consultas estáticas não podem ser editadas.

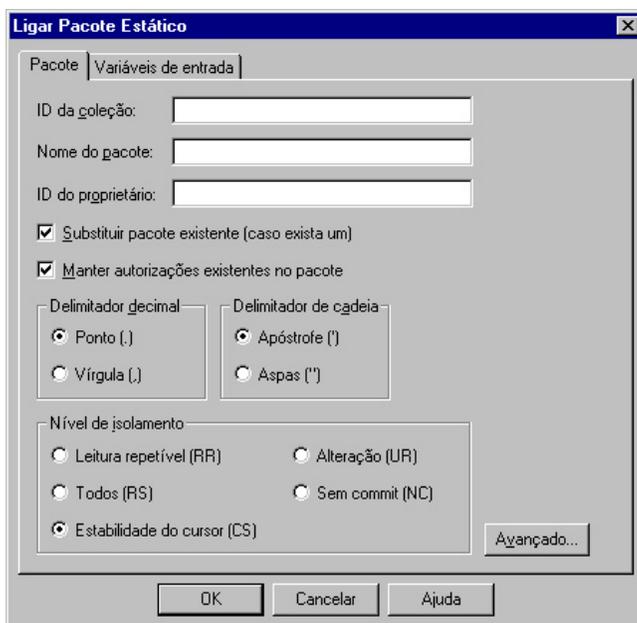
---

### Consultas Estáticas

As consultas estáticas são criadas a partir de consultas SQL e orientadas existentes.

#### Criação de consultas estáticas

1. No menu **Consulta**, selecione **Ligar Pacote Estático**. A caixa de diálogo Ligar Pacote Estático é aberta.



2. Selecione o item Pacote, digite uma ID da coleção e nome do pacote e altere quaisquer opções desejadas.
3. Se a consulta contiver quaisquer variáveis de substituição, selecione o item Variáveis. Substitua quaisquer variáveis de substituição com variáveis do host.

4. Clique em **OK**. A consulta estática é ligada.

**Nota:** Uma vez que você ligou uma consulta, você deve também salvar essa consulta ou em um arquivo ou no servidor de banco de dados.

### **Substituição de variáveis de substituição com variáveis do host**

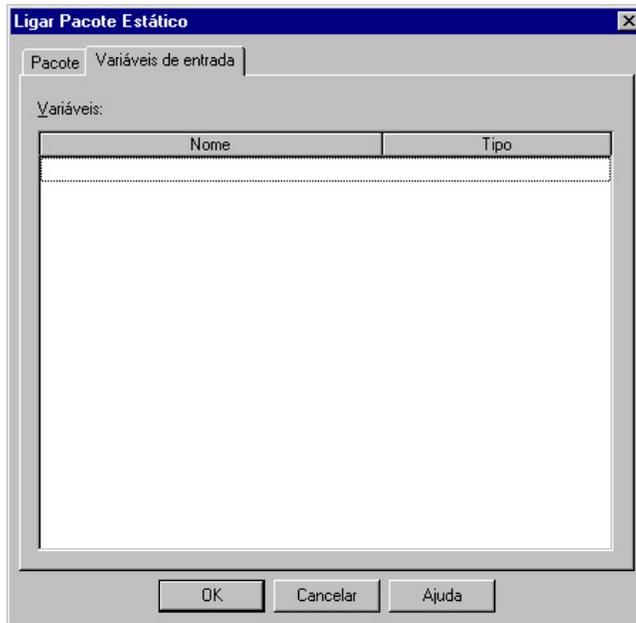
Quando você liga um pacote, você deve especificar uma variável do host para usar no lugar de cada variável de substituição no texto SQL. No entanto, essa variável não pode ser sempre substituída diretamente pela variável do host. As variáveis de substituição são substituídas diretamente no texto da consulta antes dele ser enviado ao servidor de banco de dados. Elas são enviadas como parte da consulta ao servidor de banco de dados. Consulte a documentação do servidor de banco de dados para saber as regras de onde e quando as variáveis do host podem ser usadas em consultas.

Uma vez que você especificou uma relação entre uma variável de substituição e uma variável do host, o QMF para Windows lembra-se da relação e, sugere ela nas consultas futuras, tornando a ligação de pacotes mais simples.

Os tipos de dados válidos para variáveis do host são:

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

1. A partir da caixa de diálogo Ligar Pacote Estático, selecione o item Variáveis de Entrada.



2. Digite o tipo de variável para cada variável do host e clique em **OK**. As variáveis de substituição são convertidas para variáveis do host.

### **Execução de uma consulta estática**

As consultas estáticas são executadas da mesma forma que qualquer outra consulta seria. Consulte "Consultas SQL" na página "Consultas SQL" na página 9.



---

## Capítulo 10. Operação com o Editor de Tabelas

Use o editor de tabelas para procurar, incluir, editar ou eliminar dados armazenados nas suas tabelas sem ter que escrever instruções SQL.

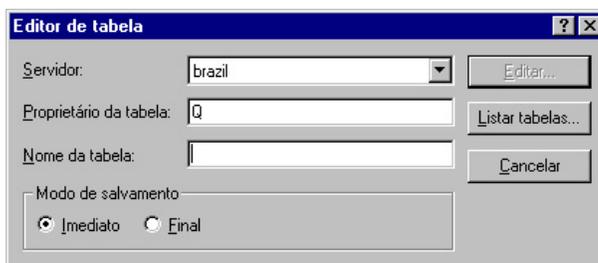
---

### Editor de tabela

O editor de tabela dá flexibilidade para edição e pesquisa de dados.

#### Pesquisa de linhas usando o editor de tabela

1. No menu **Arquivo**, selecione **Editor de Tabela**. A caixa de diálogo Editor de Tabela é aberta.



2. Especifique uma tabela.

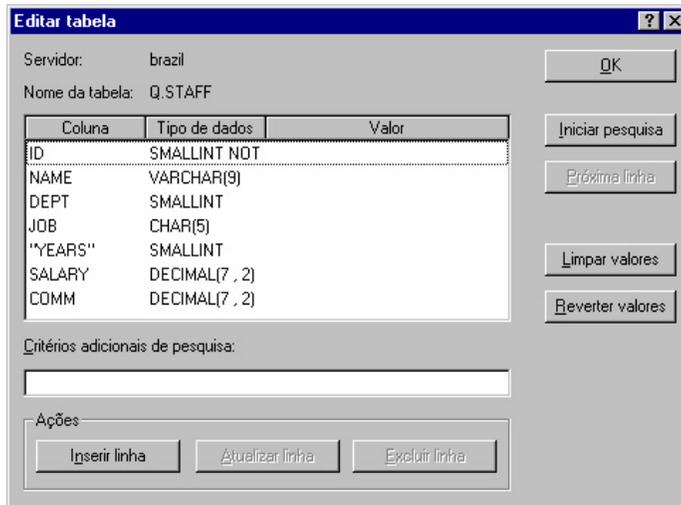
**Nota:** Você pode usar padrões para selecionar nomes de tabelas a partir de uma lista de tabelas correspondentes.

- Use o caractere de porcentagem (%) para representar cadeia de qualquer tamanho que contenha qualquer caractere. Por exemplo, para listar todas as tabelas com um nome começando com a letra A, digite A%.
- Use o caractere de sublinhado (\_) para representar um caractere individual. Por exemplo, para listar todas as tabelas com um proprietário que possui a letra A na segunda posição, você digita \_A%.

Depois de fornecer um padrão, clique em **Listar Tabelas** e selecione uma tabela da lista resultante.

3. Selecione um Modo de Salvamento.
  - **Imediato** - A tabela é atualizada no servidor de banco de dados imediatamente após cada alteração.
  - **Final** - A tabela é atualizada no servidor de banco de dados após a inserção de todas as alterações. Os outros usuários não conseguirão

alterar a tabela enquanto você estiver fazendo as suas modificações.



4. Clique em **Editar**. A caixa de diálogo Editar tabela é aberta.
5. Digite os valores a serem pesquisados na coluna Valores ou digite os critérios de pesquisa no campo Critérios adicionais de pesquisa para especificar condições mais complexas. Esse campo aceita qualquer predicado SQL válido.
6. Clique em **Iniciar Pesquisa**. A primeira linha com correspondente é exibida na coluna Valor.

### Inclusão de uma linha

1. Na caixa de diálogo Editar tabela, forneça as informações para o novo registro.
2. Clique em **Inserir Linha**. A nova linha é adicionada à tabela.
3. Clique em **OK**. Suas alterações são salvas.

### Alteração de linha

1. A partir da caixa de diálogo Editar tabela, procure a linha que deseja alterar.
2. Clique em **Próxima Linha** até que a linha que você deseja alterar seja exibida.
3. Edite os dados na coluna Valor e clique em **Atualizar Linha**. A linha é atualizada.
4. Clique em **OK**. Suas alterações são salvas.

### Eliminação de uma linha

1. A partir da caixa de diálogo Editar tabela, procure a linha que deseja eliminar.

2. Clique em **Próxima Linha** até que a linha que você deseja eliminar seja exibida.
3. Clique em **Eliminar Linha**. A linha é eliminada.
4. Clique em **OK**. Suas alterações são salvas.

---

## **Edição de Tabelas a Partir da Exibição dos Resultados da Consulta**

As tabelas podem ser editadas diretamente a partir da exibição dos resultados da consulta.

### **Eliminação de uma linha da exibição dos resultados da consulta**

Linhas individuais podem ser eliminadas a partir de tabelas na exibição de resultados da consulta.

A partir da exibição dos resultados da consulta, selecione uma linha e selecione **Eliminar** no menu **Editar**. A linha é eliminada.

### **Atualização de colunas a partir da exibição dos resultados da consulta**

O conteúdo de colunas individuais pode ser atualizado na exibição dos resultados da consulta.

A partir da exibição dos resultados da consulta, dê um clique duplo sobre a célula, forneça um novo valor e pressione Enter. A tabela é atualizada.

---

## **DB2 Forms**

Se você tiver o componente DB2 Forms User instalado em sua máquina, pode utilizá-lo como editor de tabela para tabelas que não contenham dados de LOB. Para obter mais informações sobre o DB2 Forms, visite o Centro de Recursos do DB2 Forms em [www.rocketsoftware.com/db2forms](http://www.rocketsoftware.com/db2forms).



---

## Capítulo 11. Distribuição de Dados

Seus dados podem ser exportados para outros bancos de dados e aplicações.

---

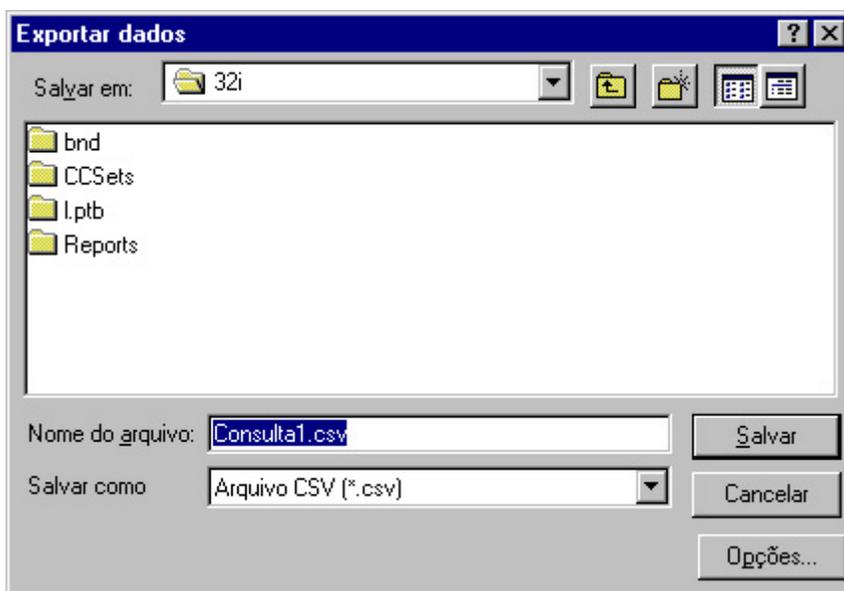
### Exportação de dados

Você pode exportar dados do QMF para Windows para outras aplicações das seguintes formas:

- Exportar os dados para um arquivo de Texto, CSV, IXF ou HTML
- Salvar os resultados da consulta em uma tabela
- Incluir os resultados da consulta diretamente em uma planilha do Microsoft Excel

### Exportação de dados para arquivos

1. Enquanto visualiza os resultados da consulta, selecione **Exportar Dados** no menu **Arquivo**. A caixa de diálogo Exportar Dados é aberta.



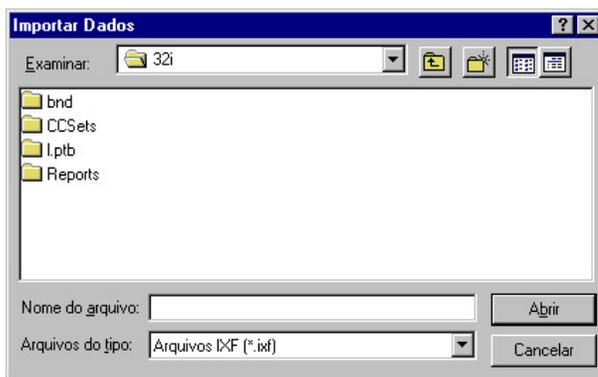
2. Selecione o Tipo de Arquivo de Saída desejado e clique no botão **Opções**. Dependendo do tipo do arquivo de saída selecionado, a caixa de diálogo Opções de Exportação de Texto/DEL, Opções de Exportação de HTML, Opções de Exportação de IXF ou Opções de Exportação de CSV é aberta.

- Você pode produzir um arquivo de texto com uma extensão .TXT. Este é um arquivo ASCII padrão com delimitadores opcionais de cadeia e coluna (como especificado na caixa de diálogo Opções de Exportação Texto/DEL).
  - Você pode produzir um arquivo HTML com uma extensão de arquivo .HTM. Esse é um arquivo HTML que pode ser exibido por qualquer navegador de rede. Todas as marcações HTML são automaticamente geradas no arquivo; ele está pronto para ser publicado no seu site da Internet ou da sua intranet. As opções escolhidas na caixa de diálogo Opções de Exportação de HTML controlam a aparência dos dados exportados.
  - Você pode produzir um arquivo .IXF. Uma exportação IXF preserva todas as informações do banco de dados, incluindo títulos de coluna e tipos de dados. É utilizado geralmente para transferir informações de um banco de dados para outro.
  - Você pode produzir um arquivo .CSV. Uma exportação CSV é bastante semelhante a uma exportação de texto, utilizando uma vírgula como delimitador de coluna. Este formato é mais comumente utilizado por aplicações de planilha.
3. Selecione as opções para o tipo selecionado de arquivo de exportação e clique em **OK**. A caixa de diálogo Opções é fechada.
  4. Clique em **OK** na caixa de diálogo Exportar Dados. Os dados são exportados.

## Importação de dados

Você pode importar dados que foram salvos em um arquivo IXF. Depois que os dados foram importados para uma janela de consulta, eles podem ser salvos em um servidor de banco de dados, exportados para um novo arquivo ou utilizados para relatórios. Os arquivos PC/IXF e os arquivos IXF de modo de caractere System/370 são suportados.

1. No menu **Arquivo**, selecione **Importar Dados**. A caixa de diálogo Importar Dados é aberta.

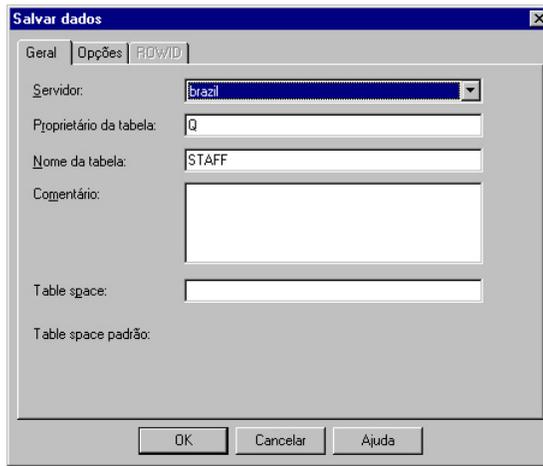


2. Selecione o arquivo que deseja importar e clique em **OK**. Os dados importados são exibidos em uma nova janela de consulta.

## Salvamento de dados em um servidor de banco de dados

Você pode salvar resultados da consulta importados em uma tabela do banco de dados.

1. Enquanto visualiza os resultados da consulta importados, selecione **Salvar Dados** no menu **Arquivo**. A caixa de diálogo Salvar Dados é aberta.



2. Selecione um servidor de banco de dados, digite um proprietário e nome de tabela, escolha quaisquer outras opções desejadas e clique em **OK**. Os dados são salvos.

---

## Uso do comando Enviar Para

O QMF para Windows inclui um comando Enviar Para e um cliente de e-mail básico. Você pode utilizar o comando Enviar Para em conjunto com arquivos de job para programar consultas e distribuir seus resultados.

1. No menu **Arquivo**, selecione **Enviar Para** e **Destinatário de Correio da Internet**. A caixa de diálogo Mensagem é aberta.
2. Especifique um destinatário da mensagem, um assunto, o texto da mensagem e clique em **Avançar**. A caixa de diálogo Anexos é aberta.
3. Inclua ou remova quaisquer anexos da mensagem e clique em **Avançar**. A caixa de diálogo Enviar Mensagem é aberta.
4. Especifique o nome do servidor de correio e clique em **Concluir**. A mensagem é enviada.

---

## Uso do Complemento Microsoft Excel

O QMF para Windows inclui um complemento para o Microsoft Excel 7.0 ou posterior. Estes complementos permitem executar o QMF para Windows a partir do Excel e os resultados da consulta podem ser enviados diretamente para uma planilha. O complemento apropriado é instalado automaticamente quando se escolhe a opção de instalação "Típica" ou se escolhe a opção de instalação "Personalizada" e se seleciona a opção do Complemento Microsoft Excel.

1. Clique no botão **QMF para Windows** na barra de ferramentas do Excel.



O QMF para Windows é aberto.

2. A partir do QMF para Windows, selecione e execute uma consulta. Os resultados da consulta aparecem.
3. Selecione os dados que deseja retornar ao Excel.
4. No menu **Arquivo**, selecione **Retornar dados para o Microsoft Excel**. O Excel abre e exibe a caixa de diálogo Complemento do QMF para Windows.
5. Digite o intervalo de destino para os dados e clique em **OK**. Os dados são incluídos na planilha.

---

## Uso das Aplicações de Exemplo

Várias aplicações de exemplo e soluções de integração estão disponíveis para o QMF para Windows. Visite o site web da IBM no endereço <http://www.ibm.com/qmf/> para obter mais informações.

---

## Capítulo 12. Uso da Central de Relatórios do QMF

A Central de Relatórios do QMF permite produzir relatórios personalizados utilizando consultas, formulários, procedimentos e tabelas compartilhados do QMF. Com acesso rápido a esses objetos, você pode especificar preferências de formato de dados e produzir relatórios personalizados que podem ser visualizados e manipulados em diversos aplicativos.

---

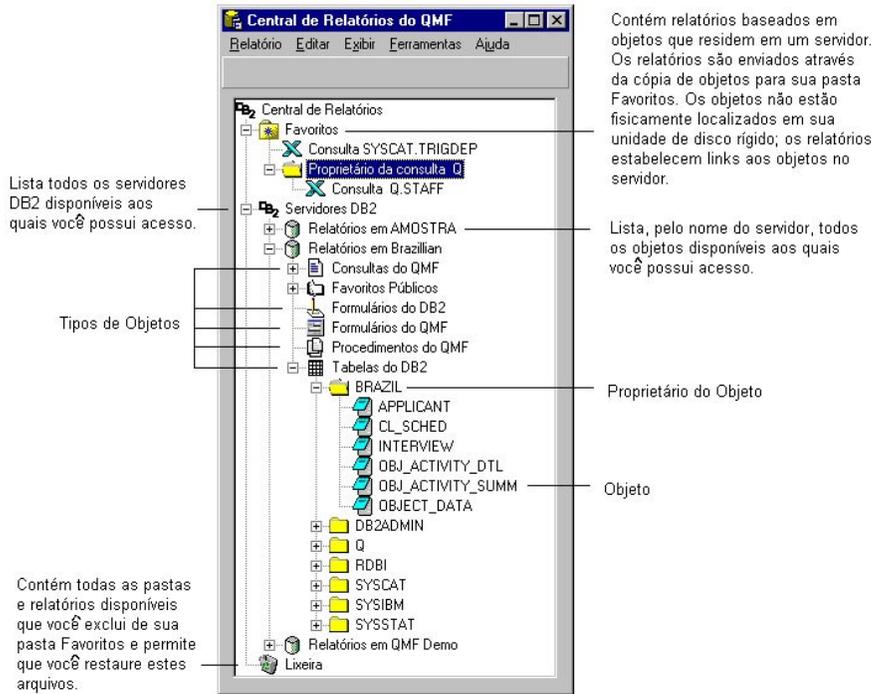
### Introdução à Central de Relatórios do QMF

- Clique com o botão direito em qualquer objeto ou pasta para ativar as mesmas opções que estão disponíveis nos menus da barra de ferramentas.
- Clique no sinal de mais (+) ao lado de qualquer pasta para abrir o primeiro nível de conteúdo. Mantenha a tecla SHIFT pressionada enquanto clica no sinal de mais (+) para abrir todos os níveis abaixo da pasta.

### Janela da Central de Relatórios do QMF

A janela da Central de Relatórios do QMF contém uma estrutura em árvore de Favoritos, servidores do DB2, Favoritos Públicos, objetos disponíveis e uma

Lixeira.

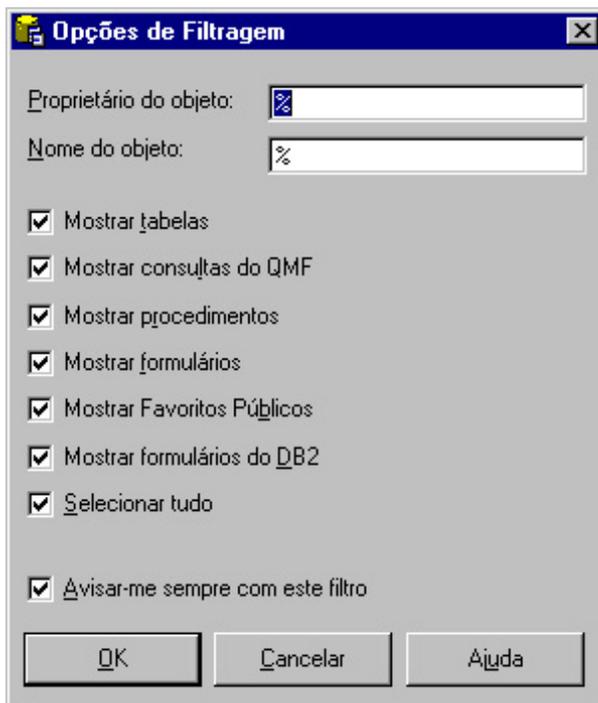


Os objetos exibidos nesta janela incluem um ícone que representa o tipo de aplicativo ao qual a saída do objeto está associada.

## Conexão ao Servidor

1. Se não forem exibidos nomes de servidor sob os Servidores do DB2, clique no sinal de mais (+).

2. Clique no sinal de mais (+) ao lado de um servidor. A caixa de diálogo Opções de Filtragem é aberta.



3. Selecione os tipos de objeto que deseja visualizar e clique em **OK**. Os objetos disponíveis no servidor são exibidos, agrupados por tipo de objeto.

---

## Operação com Relatórios e Objetos

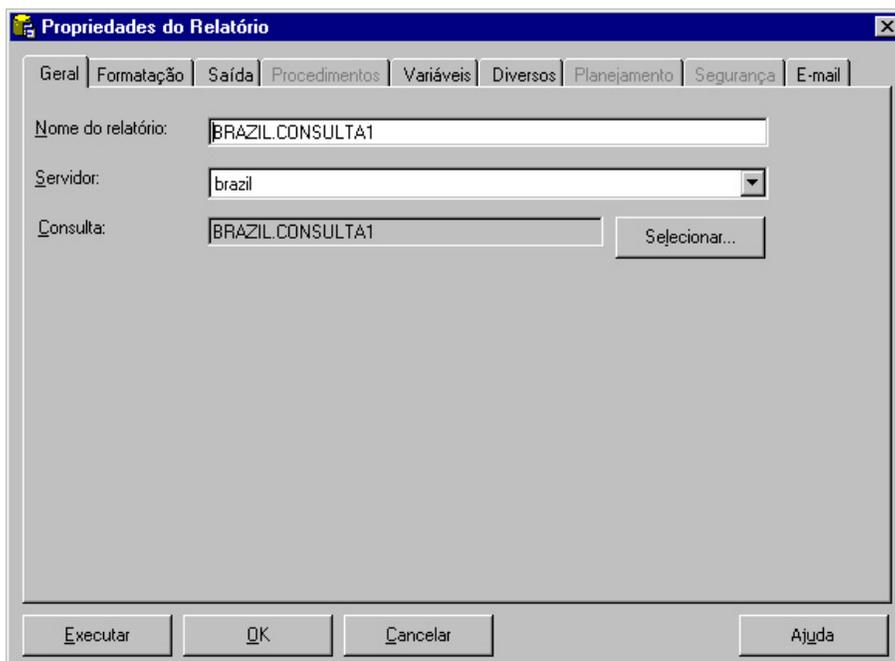
Relatórios são baseados em objetos do QMF. Todos os itens contidos em suas pastas pessoal e Favoritos Públicos são considerados relatórios; você pode manipular a formatação e as opções de exibição para esses itens. Os itens contidos nessas pastas Favoritos são ligados a objetos do QMF que residem no servidor. Na verdade, você não modifica um objeto do QMF, modifica o link para o objeto que é referenciado como um relatório. Como os relatórios são baseados em objetos, as propriedades dos objetos também se aplicam a relatórios.

Você pode criar relatórios a partir de objetos que residem em um servidor; porém eles não são salvos no servidor. Esta funcionalidade permite criar rapidamente relatórios para usar uma só vez. Entretanto, após criar um relatório a partir de objetos em um servidor, você recebe a opção de salvá-lo em sua pasta Favoritos.

## Execução de Relatórios

Você pode executar relatórios a partir de sua pasta Favoritos ou a partir de objetos localizados no servidor.

1. Com o relatório ou objeto selecionado, escolha *Propriedades* no menu Relatório. A caixa de diálogo Propriedades do Relatório é aberta.



2. Defina as propriedades, se desejar.
3. Clique no botão **Executar**. O relatório é processado e, se você selecionou a opção *Exibir relatório após publicação* na caixa de diálogo Saída de Propriedades do Relatório, o relatório é exibido no aplicativo especificado.

Você também pode executar rapidamente um relatório através de qualquer um dos seguintes métodos:

- Selecione o relatório e escolha *Executar*, no menu Relatório.
- Clique com o botão direito no relatório e escolha *Executar*.
- Dê um clique duplo no nome do relatório.

---

## Operação com Pastas e Favoritos

Pastas são utilizadas para agrupar relatórios e objetos do QMF; elas são nomeadas de acordo com os nomes dos proprietários dos objetos. Você pode realizar nas pastas as mesmas operações que realiza nos relatórios, como executar relatórios e definir propriedades de relatórios. Ao executar essas

operações em uma pasta faz com que a operação seja aplicada a cada relatório contido na pasta. Por exemplo, se você desejar executar consecutivamente cada relatório contido em uma pasta, selecione a pasta e, em seguida, selecione **Executar** no menu Relatório.

A Central de Relatórios do QMF contém duas pastas de nível superior nas quais você pode armazenar relatórios. As pastas contêm relatórios que apontam para objetos em servidores; os objetos em si não estão contidos nas pastas Favoritos. A sua pasta Favoritos pessoal reside localmente (em seu PC), portanto você é o único usuário que pode acessar a pasta e seu conteúdo. A pasta Favoritos Públicos reside no servidor e pode ser acessada por todos os usuários autorizados. Você pode ter acesso a várias pastas Favoritos Públicos, dependendo dos limites dos seus recursos, embora nunca haja mais de uma pasta Favoritos Públicos em cada servidor.

Quando você copia objetos do QMF para as pastas Favoritos, estas são automaticamente renomeadas de modo a incluir o tipo de objeto e o nome do proprietário. Quando você copia um tipo de objeto inteiro (ou seja, pasta de objetos do mesmo tipo) de um servidor (por exemplo, todas as consultas), o nome do servidor também é incluído no nome da nova pasta.

### **Adição de Relatórios a Favoritos**

Você pode incluir qualquer objeto ou relatório de um servidor à sua pasta Favoritos pessoal ou à pasta Favoritos Públicos em um servidor (desde que tenha recebido permissão do administrador do sistema).

#### **Para incluir relatórios a Favoritos pessoais:**

Com o relatório ou objeto selecionado, escolha *Incluir em favoritos* no menu Relatório ou arraste o relatório ou objeto para a pasta Favoritos pessoal. Um relatório é incluído no início da pasta Favoritos pessoal com a seguinte convenção de nomenclatura:

TipobjetoNOMEPROPRIETÁRIO.NOMEOBJETO.

#### **Para incluir relatórios em Favoritos Públicos:**

Arraste o objeto ou relatório do QMF para a pasta Favoritos Públicos em um servidor. Você pode incluir relatórios de sua pasta Favoritos pessoal ou de qualquer servidor.

**Nota:** Ao incluir Favoritos Públicos ou modificar relatórios em Favoritos Públicos, você deve selecionar *Salvar alterações em Favoritos Públicos* no menu Relatório para que suas atualizações sejam salvas no servidor.

Para obter mais informações sobre a utilização da Central de Relatórios do QMF, consulte o sistema de ajuda online.



---

## Capítulo 13. Uso da API do QMF para Windows

Você pode criar aplicativos personalizados usando a interface de programação de aplicativos do QMF para Windows.

---

### Controle do QMF para Windows pela API

As etapas a seguir oferecem uma visão geral de como trabalhar com a API para controlar o QMF para Windows.

1. Crie uma instância do objeto API do QMF para Windows. Se estiver utilizando o Microsoft Visual Basic, inclua uma referência na biblioteca de tipos do QMF para Windows, qmfwin.tlb. Em seguida, use a instrução Dim:

```
Dim QMFWin As New QMFWin
```

Ou a instrução CreateObject:

```
Dim QMFWin As Object
```

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

**Nota:** Se estiver utilizando um ambiente de desenvolvimento diferente, consulte a documentação do produto sobre como concluir esta etapa.

2. Selecione o servidor DB2 que deseja utilizar e chame InitializeServer() para inicializar uma conexão com o banco de dados.

**Nota:** Não é possível inicializar um servidor até que um ID do usuário e senha sejam validados pelo DB2. Você pode instruir o QMF para Windows para solicitar o ID do usuário e a senha, ou pode solicitá-los em seu aplicativo e passá-los como parâmetros na chamada de função InitializeServer().

3. Selecione a consulta que deseja executar utilizando InitializeQuery(). Se a consulta contiver variáveis, utilize a função SetVariable() para definir os valores das variáveis.
4. Abra ou execute a consulta. Utilize a função Open() para abrir o cursor da consulta para instruções SELECT e utilize a função Execute() para executar o SQL para instruções não SELECT.
5. Se a consulta for uma instrução SELECT, busque linhas de dados chamando repetidamente FetchNextRow(). Para buscar mais de uma linha por vez, utilize FetchNextRows() ou utilize CompleteQuery() para direcionar o QMF para Windows para que busque todas as linhas.

6. Se a consulta for uma instrução SELECT, feche-a utilizando a função Close().
7. Termine a unidade de trabalho utilizando as funções Commit() ou Rollback().

## **Bloqueio de chamadas**

Todas as funções da API do QMF para Windows são síncronas. Isto significa que elas bloqueiam, ou não retornam, até que a ação solicitada do banco de dados seja concluída. Esta implementação é desejável porque simplifica a programação no aplicativo cliente. Entretanto, se o aplicativo cliente for de encadeamento único, ele não pode responder à entrada do usuário ou executar atualizações de tela enquanto aguarda que uma função da API do QMF para Windows retorne.

A API do QMF para Windows responde a uma chamada de função por vez de um cliente. Se o aplicativo cliente for de encadeamento múltiplo, você deve:

- aguardar a conclusão de uma chamada de função antes de fazer outra ou
- criar múltiplas instâncias da API do QMF para Windows (uma para cada encadeamento que utilizar a API).

## **Conexão ao banco de dados**

Cada instância do objeto da API do QMF para Windows cria e utiliza uma única conexão ao banco de dados para todas as atividades do banco de dados sujeitas a uma reversão ou consolidação posterior, incluindo a abertura de uma consulta, busca de dados e execução de instruções SQL.

Se você criar mais de uma consulta em uma determinada instância do objeto da API do QMF para Windows chamando InitializeQuery() duas ou mais vezes, todas as consultas compartilharão a mesma conexão única.

A API do QMF para Windows responde a uma chamada de função por vez de um cliente. Se o aplicativo cliente for de encadeamento múltiplo, você deve:

- DeleteQMFObject()
- GetQMFOBJECTInfo()
- GetQMFOBJECTInfoEx()
- GetQMFOBJECTList()
- GetQMFOBJECTListEx()
- GetQMFOBJECTQueryText()
- SaveQMFOBJECTQuery()

O QMF para Windows cria e utiliza uma segunda conexão ao banco de dados para tratar atividades administrativas do banco de dados (por exemplo,

recuperação de informações do QMF). Esta segunda conexão é necessária para suportar um mecanismo consistente de reversão e consolidação para aplicativos clientes.

O objeto da API do QMF para Windows trata automaticamente estas conexões com o banco de dados. Entretanto, se o administrador do sistema tiver estabelecido um limite para o número de conexões permitidas, lembre-se que cada instância do objeto da API do QMF para Windows pode utilizar duas conexões.

---

## Referência da API

Esta referência lista todos os comandos disponíveis para criar aplicativos utilizando a API do QMF para Windows.

### AddDecimalHostVariable()

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

#### Descrição

Esta função aplica os dados em *Valor* a uma variável na instrução SQL estática inicializada com *IDConsulta*. Chame esta função para cada variável na instrução. O QMF para Windows não tenta coincidir valores com variáveis, portanto você deve chamar esta função na mesma ordem que as variáveis na instrução SQL.

#### Parâmetros

Nome	Descrição
<i>IDConsulta</i>	O ID da consulta, conforme retornado de <code>InitializeStaticQuery()</code> .
<i>Tipo</i>	O tipo de dados SQL do valor a ser passado para o servidor do banco de dados. Este valor influencia a conversão de <i>Valor</i> a partir de um tipo de dados VARIANT para o valor efetivamente passado. O único valor válido para <code>AddDecimalHostVariable()</code> é 484 (RSDT_DECIMAL).
<i>Precisão</i>	A precisão do valor decimal.
<i>Escala</i>	A escala do valor decimal.
<i>Valor</i>	O valor dos dados a substituir na instrução. Para especificar um valor nulo, o tipo da variante deve ser definido como VT_EMPTY.

#### Valor de Retorno

Zero se tiver êxito, diferente de zero se não tiver êxito. Se o valor de retorno for diferente de zero, você pode chamar `GetLastErrorString()` ou `GetLastErrorType()` para obter informações de erro adicionais.

## AddHostVariable()

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

### Descrição

Esta função aplica os dados em *Valor* a uma variável na instrução SQL estática inicializada com *IDConsulta*. Você deve chamar esta função para cada variável na instrução. O QMF para Windows não tenta coincidir valores com variáveis, portanto você deve chamar esta função na mesma ordem que as variáveis na instrução SQL.

### Parâmetros

Nome	Descrição
<i>IDConsulta</i>	O ID da consulta, conforme retornado de InitializeStaticQuery().
<i>Tipo</i>	O tipo de dados SQL do valor a ser passado para o servidor do banco de dados. Este valor influencia a conversão de <i>Valor</i> a partir de um tipo de dados VARIANT para o valor efetivamente passado.
<i>Valor</i>	O valor dos dados a substituir na instrução. Para especificar um valor nulo, o tipo da variante deve ser definido como VT_EMPTY.

Os valores válidos para *Tipo* incluem:

Valor	Significado
384 (RSDT_DATE)	Data
388 (RSDT_TIME)	Hora
392 (RSDT_TIMESTAMP)	Marca de hora
448 (RSDT_VARCHAR)	Cadeia de caracteres de comprimento variável
452 (RSDT_CHAR)	Cadeia de caracteres
464 (RSDT_VARGRAPHIC)	Gráfico de comprimento variável
468 (RSDT_GRAPHIC)	Gráfico
480 (RSDT_FLOAT)	Número de ponto flutuante
496 (RSDT_INTEGER)	Inteiro de 4 bytes
500 (RSDT_SMALLINT)	Inteiro de 2 bytes

### Valor de Retorno

Zero se tiver êxito, diferente de zero se não tiver êxito. Se o valor de retorno for diferente de zero, você pode chamar GetLastErrorString() ou GetLastErrorType() para obter informações de erro adicionais.

## BindDecimalHostVariable()

short BindDecimalHostVariable(BSTR *NomeColeção*, BSTR *NomePacote*, short *NúmeroSeção*, short *Número*, BSTR *Nome*, short *TipoDados*, short *Precisão*, short *Escala*)

### Descrição

Esta função liga uma variável na seção especificada. Inclua o texto ":H" no texto SQL como espaço reservado para uma variável de host. Para cada variável de host decimal no texto SQL, você deve chamar BindDecimalHostVariable() para especificar informações sobre a mesma.

### Parâmetros

Nome	Descrição
NomeColeção	O ID da coleção do pacote que você deseja ligar.
NomePacote	O nome do pacote que você deseja ligar.
NúmeroSeção	O número da seção da instrução dentro da coleção e pacote que você deseja ligar.
Número	O identificador da variável que você deseja ligar. A primeira variável na instrução SQL é a variável 0, etc.
Nome	Utilizado pelo servidor do banco de dados para fins de diagnóstico. Este valor não é validado nem requerido pelo QMF para Windows.
TipoDados	O tipo de dados SQL da variável. O único valor válido para BindDecimalHostVariable() é 484 (RSDT_DECIMAL).
Precisão	A precisão do valor decimal.
Escala	A escala do valor decimal.

### Valor de Retorno

Zero se tiver êxito, diferente de zero se não tiver êxito. Se o valor de retorno não for zero, você pode chamar GetLastErrorString() ou GetLastErrorType(), GetLastErrorSQLCode(), GetLastErrorSQLError() ou GetLastErrorSQLState() para obter informações de erro adicionais.

## BindHostVariable()

short BindHostVariable(BSTR *NomeColeção*, BSTR *NomePacote*, short *NúmeroSeção*, short *Número*, BSTR *Nome*, short *TipoDados*, short *Comprimento*)

### Descrição

Esta função liga uma variável na seção especificada. Inclua o texto ":H" no texto SQL como espaço reservado para uma variável de host. Para cada variável de host no texto SQL, você deve chamar BindHostVariable() para especificar informações sobre a variável.

## Parâmetros

Nome	Descrição
<i>NomeColeção</i>	O ID da coleção do pacote que você deseja ligar.
<i>NomePacote</i>	O nome do pacote que você deseja ligar.
<i>NúmeroSeção</i>	O número da seção da instrução dentro da coleção e pacote que você deseja ligar.
<i>Número</i>	O identificador da variável que você deseja ligar. A primeira variável na instrução SQL é a variável 0, etc.
<i>Nome</i>	Utilizado pelo servidor do banco de dados para fins de diagnóstico. Este valor não é validado nem requerido pelo QMF para Windows.
<i>TipoDados</i>	O tipo de dados SQL da variável.
<i>Comprimento</i>	O comprimento da variável.

Os valores válidos para *TipoDados* incluem:

Valor	Significado
384 (RSDT_DATE)	Data
388 (RSDT_TIME)	Hora
392 (RSDT_TIMESTAMP)	Marca de hora
448 (RSDT_VARCHAR)	Cadeia de caracteres de comprimento variável
452 (RSDT_CHAR)	Cadeia de caracteres
464 (RSDT_VARGRAPHIC)	Gráfico de comprimento variável
468 (RSDT_GRAPHIC)	Gráfico
480 (RSDT_FLOAT)	Número de ponto flutuante
484 (RSDT_DECIMAL)	Decimal
496 (RSDT_INTEGER)	Inteiro de 4 bytes
500 (RSDT_SMALLINT)	Inteiro de 2 bytes

### Valor de Retorno

Zero se tiver êxito, diferente de zero se não tiver êxito. Se o valor de retorno não for zero, você pode chamar `GetLastErrorString()` ou `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` ou `GetLastSQLState()` para obter informações de erro adicionais.

### BindSection()

```
short BindSection(BSTR NomeColeção, BSTR NomePacote, short NúmeroSeção,  
BSTR TextoSQL)
```

### Descrição

Esta função define o texto SQL a ser utilizado no número da seção especificado da coleção e pacote durante a ligação.

### Parâmetros

Nome	Descrição
NomeColeção	O ID da coleção do pacote que você deseja ligar.
NomePacote	O nome do pacote que você deseja ligar.
NúmeroSeção	O número da seção da instrução dentro da coleção e pacote que você deseja ligar.
<i>TextoSQL</i>	O texto SQL para a instrução que você deseja ligar.

### Valor de Retorno

Zero se tiver êxito, diferente de zero se não tiver êxito. Se o valor de retorno não for zero, você pode chamar `GetLastErrorString()` ou `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` ou `GetLastSQLState()` para obter informações de erro adicionais.

## CancelBind()

short `CancelBind(BSTR NomeColeção, BSTR NomePacote)`

### Descrição

Esta função cancela uma operação de ligação anteriormente inicializada. Todas as informações sobre o pacote nomeado são liberadas.

### Parâmetros

Nome	Descrição
NomeColeção	O ID da coleção do pacote que você deseja ligar.
NomePacote	O nome do pacote que você deseja ligar.

### Valor de Retorno

Zero se tiver êxito, diferente de zero se não tiver êxito. Se o valor de retorno não for zero, você pode chamar `GetLastErrorString()` ou `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` ou `GetLastSQLState()` para obter informações de erro adicionais.

## ChangePassword()

short `ChangePassword(BSTR NovaSenha)`

### Descrição

Esta função altera a senha para o ID do usuário anteriormente especificado na chamada `InitializeServer()`.

**Nota:** Nem todos os tipos de servidores de banco de dados suportam a alteração de senhas. Se o servidor especificado na chamada `InitializeServer()` não suportar a alteração de senhas, um erro é retornado e a senha não é alterada.

### Parâmetros

Nome	Descrição
NovaSenha	A nova senha.

### Valor de Retorno

Zero se tiver êxito, diferente de zero se não tiver êxito. Se o valor de retorno não for zero, você pode chamar `GetLastErrorString()` ou `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` ou `GetLastSQLState()` para obter informações de erro adicionais.

## ClearList()

short `ClearList(short Tipo)`

### Descrição

Esta função reinicializa a lista interna especificada pelo parâmetro *Tipo*.

### Parâmetros

Nome	Descrição
<i>Tipo</i>	Ou o valor <code>RSL_SERVER</code> ou <code>RSL_QUERY</code> .

### Valor de Retorno

Zero se tiver êxito, `RS_ERROR_OUTOFRANGE` se não tiver êxito.

### Tópicos Relacionados

`Open()`

## Close()

short `Close(long IDConsulta)`

### Descrição

Esta função fecha uma consulta e invalida *IDConsulta*. Se houver um cursor aberto para a consulta, este é fechado, liberando o banco de dados para outros usuários. Esta função não termina a conexão com o servidor do banco de dados. Como a conexão permanece aberta, não é realizada reversão ou consolidação.

**Nota:** O nome desta função entra em conflito com a palavra-chave `Close` do Microsoft Access 2.0. Se você estiver utilizando o MS Access 2.0, coloque o nome da função entre colchetes [ ].

## Parâmetros

Nome	Descrição
IDConsulta	O ID da consulta, conforme retornado de InitializeQuery().

### Valor de Retorno

Zero se tiver êxito, diferente de zero se não tiver êxito. Se o valor de retorno não for zero, você pode chamar GetLastErrorString() ou GetLastErrorType(), GetLastErrorSQLCode(), GetLastErrorSQLError() ou GetLastErrorSQLState() para obter informações de erro adicionais.

### Tópicos Relacionados

Execute()

Open()

## Commit()

short Close(long *IDConsulta*)

### Descrição

Esta função consolida quaisquer alterações feitas na unidade de trabalho atual, termina a unidade de trabalho atual, fecha todos os cursores abertos e invalida todos os IDs de Consultas pendentes.

**Nota:** O nome desta função entra em conflito com a palavra-chave Commit do Microsoft Access 2.0. Se você estiver utilizando o MS Access 2.0, coloque o nome da função entre colchetes [ ].

### Valor de Retorno

Zero se tiver êxito, diferente de zero se não tiver êxito. Se o valor de retorno não for zero, você pode chamar GetLastErrorString() ou GetLastErrorType(), GetLastErrorSQLCode(), GetLastErrorSQLError() ou GetLastErrorSQLState() para obter informações de erro adicionais.

### Tópicos Relacionados

Rollback()

## CompleteQuery()

short CompleteQuery(long *IDConsulta*)

### Descrição

Esta função busca todas as linhas de um conjunto de resultados e armazena-as internamente no QMF para Windows. Se houver um cursor aberto para a consulta, este é fechado, liberando o banco de dados para outros usuários. Você pode utilizar FetchNextRow() ou FetchNextRows() para recuperar as linhas. Chame Close() quando tiver terminado esta consulta.

## Parâmetros

Nome	Descrição
<i>IDConsulta</i>	O ID da consulta, conforme retornado de <code>InitializeQuery()</code> .

### Valor de Retorno

Zero se tiver êxito, diferente de zero se não tiver êxito. Se o valor de retorno não for zero, você pode chamar `GetLastErrorString()` ou `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` ou `GetLastSQLState()` para obter informações de erro adicionais.

## CopyToClipboard()

`short CopyToClipboard(long IDConsulta, long PrimeiraLinha, long PrimeiraCol, long ÚltimaLinha, long ÚltimaCol, BOOL IncluirTítulosCol, [VARIANT FormatoDataHora])`

### Descrição

Esta função copia o intervalo de linhas e colunas especificado para a Área de transferência. Se você não tiver recuperado dados de linhas para todas as linhas que deseja copiar para a Área de transferência, chame `CompleteQuery()` antes de chamar esta função. Uma mensagem de erro será retornada se você tentar copiar linhas que não foram recuperadas do banco de dados.

## Parâmetros

Nome	Descrição
<i>IDConsulta</i>	O ID da consulta, conforme retornado de <code>InitializeQuery()</code> .
<i>PrimeiraLinha</i>	A primeira linha que você deseja incluir na cópia.
<i>PrimeiraCol</i>	A primeira coluna que você deseja incluir na cópia.
<i>ÚltimaLinha</i>	A última linha que você deseja incluir na cópia ou -1 se todas as linhas forem incluídas.
<i>ÚltimaCol</i>	A última coluna que você deseja incluir na cópia ou -1 se todas as colunas forem incluídas.
<i>IncluirTítulosCol</i>	Utilize um valor diferente de zero para incluir os títulos das colunas na primeira linha e zero para não inclui-los.
<i>FormatoDataHora</i>	Opcionalmente, o formato a utilizar para valores de data e hora. Os valores válidos são 0 (formato ISO), 1 (formato EUA), 2 (formato EUR), 3 (formato JIS) ou 4 (formato do Painel de controle do Windows). O valor padrão é 4.

**Nota:** O valor da primeira linha em um conjunto de resultados é 0 e o valor da última linha é um a menos que o número total de linhas. O valor da

primeira coluna em um conjunto de resultados é 0 e o valor da última coluna é um a menos que o número total de colunas.

### Valor de Retorno

Zero se tiver êxito, diferente de zero se não tiver êxito. Se o valor de retorno for diferente de zero, você pode chamar `GetLastErrorString()` ou `GetLastErrorType()` para obter informações de erro adicionais. Se o conjunto de resultados for vazio ou não tiverem sido recuperadas linhas do banco de dados, um valor diferente de zero é retornado exceto se *PrimeiraLinha*=0 e *ÚltimaLinha*=1. Neste caso, zero é retornado e uma cadeia vazia é copiada para a Área de transferência.

## DeleteQMFObject()

short DeleteQMFObject(BSTR *ProprietarioNome*)

### Descrição

Esta função exclui um objeto do QMF (consulta, formulário, procedimento ou tabela).

### Parâmetros

Nome	Descrição
<i>ProprietarioNome</i>	Uma cadeia que contém o proprietário e o nome, separados por um ponto, do objeto que você deseja excluir. Por exemplo, Paulo.Consulta2

### Valor de Retorno

Zero se tiver êxito, diferente de zero se não tiver êxito. Se o valor de retorno não for zero, você pode chamar `GetLastErrorString()` ou `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` ou `GetLastSQLState()` para obter informações de erro adicionais.

## EndBind()

short EndBind(BSTR *NomeColeção*, BSTR *NomePacote*)

### Descrição

Esta função conclui o processo de ligação para um pacote SQL estático. A chamada desta função faz com que o QMF para Windows envie as informações completas do pacote atual para o banco de dados para processamento.

### Parâmetros

Nome	Descrição
<i>NomeColeção</i>	O nome da coleção utilizada na chamada anterior a <code>StartBind()</code> .
<i>NomePacote</i>	O nome do pacote utilizado na chamada anterior a <code>StartBind()</code> .

### Valor de Retorno

Zero se tiver êxito, diferente de zero se não tiver êxito. Se o valor de retorno não for zero, você pode chamar `GetLastErrorString()` ou `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` ou `GetLastSQLState()` para obter informações de erro adicionais.

## Execute()

`short Execute(long IDConsulta)`

### Descrição

Esta função executa uma instrução SQL que utiliza um verbo SQL diferente de `SELECT`. Utilize `Execute()` quando a instrução não retornar resultados. Para instruções que retornam resultados, utilize `ExecuteEx()`. Para instruções que utilizam o verbo `SELECT`, utilize `Open()` ao invés de `Execute()` ou `ExecuteEx()`. Para determinar o verbo utilizado por uma consulta, chame `GetQueryVerb()`.

**Nota:** O nome desta função entra em conflito com a palavra-chave `Execute` do Microsoft Access 2.0. Se você estiver utilizando o MS Access 2.0, coloque o nome da função entre colchetes [ ].

### Parâmetros

Nome	Descrição
<i>IDConsulta</i>	O ID da consulta, conforme retornado de <code>InitializeQuery()</code> .

### Valor de Retorno

Zero se tiver êxito, diferente de zero se não tiver êxito. Se o valor de retorno não for zero, você pode chamar `GetLastErrorString()` ou `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` ou `GetLastSQLState()` para obter informações de erro adicionais.

### Tópicos Relacionados

`Execute()`

`Open()`

## ExecuteEx()

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

### Descrição

Esta função executa uma instrução SQL que utiliza um verbo SQL diferente de `SELECT`. Utilize `ExecuteEx()` quando a instrução retornar resultados; por exemplo, com uma instrução `SELECT INTO`. Para instruções que não retornam nenhum resultado, utilize `Execute()`. Para instruções que utilizam o verbo `SELECT`, utilize `Open()` ao invés de `Execute()` ou `ExecuteEx()`. Para determinar o verbo utilizado por uma consulta, chame `GetQueryVerb()`.

## Parâmetros

Nome	Descrição
<i>IDConsulta</i>	O ID da consulta, conforme retornado de <code>InitializeQuery()</code> .
<i>Resultado</i>	<p>Um ponteiro para uma VARIANT na qual o resultado é armazenado. O resultado é uma matriz (tipo de variante VT_ARRAY   VT_VARIANT) que contém um valor para cada coluna no resultado.</p> <p>Cada valor é especificado em seu tipo de dados nativo ou no tipo de dados da variante mais próximo. Os tipos de retorno suportados são: string (tipo de variante VT_BSTR), float (tipo de variante VT_R4), double (tipo de variante VT_R8), short (tipo de variante VT_I2), long (tipo de variante VT_I4) e binary (tipo de variante VT_UI1   VT_ARRAY).</p> <p>Você deve inicializar adequadamente a VARIANT antes de chamar esta função. O Visual Basic faz isto automaticamente. Programadores do Visual C++ devem chamar <code>VariantInit()</code>.</p>

### Valor de Retorno

Zero se tiver êxito, diferente de zero se não tiver êxito. Se o valor de retorno não for zero, você pode chamar `GetLastErrorString()` ou `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` ou `GetLastSQLState()` para obter informações de erro adicionais.

## ExecuteStored Procedure()

```
short ExecuteStoredProcedure(long IDConsulta,[VARIANT vaConsolidarOK],  
[VARIANT vaMáxConjResultados], [VARIANT vaNomesColunas], [VARIANT  
vaRótulosColunas], [VARIANT vaComentáriosColunas])
```

### Descrição

Esta função executa uma instrução SQL que utiliza o verbo CALL do SQL para executar um procedimento armazenado no servidor do banco de dados. Utilize `ExecuteStoredProcedure()` quando o procedimento armazenado não retornar resultados (ao invés de, ou além de, conjuntos de resultados). Para procedimentos armazenados que retornam resultados, utilize `ExecuteStoredProcedureEx()`.

Para inicializar um procedimento armazenado para execução com `ExecuteStoredProcedure()`, chame primeiro `InitializeQuery()` especificando uma instrução SQL que utilize a instrução CALL. O nome do procedimento armazenado deve ser especificado como um literal na instrução CALL. Quaisquer parâmetros especificados na instrução CALL (constantes ou não) são ignorados. Ao invés disso, utilize `AddHostVariable()` para especificar as variáveis de entrada e saída.

## Parâmetros

Nome	Descrição
<i>IDConsulta</i>	O ID da consulta, conforme retornado de <code>InitializeQuery()</code> . O texto SQL para a consulta deve especificar uma instrução <code>CALL</code> .
<i>vaConsolidarOK</i>	Um valor booleano opcional que especifica se o procedimento armazenado pode consolidar a unidade de trabalho ou se esta operação deve ser restrita. O valor padrão é <code>True</code> .
<i>vaMáxConjResultados</i>	Um valor numérico opcional que especifica o número máximo de conjuntos de resultados que o procedimento armazenado deve ter permissão para retornar. Especifique zero se não desejar que o procedimento armazenado retorne conjuntos de resultados ou se o servidor do banco de dados não suportar o retorno de conjuntos de resultados de procedimentos armazenados sobre DRDA.
<i>vaNomesColunas</i>	Um valor booleano opcional que especifica se o banco de dados deve retornar ou não nomes de colunas para as colunas em cada conjunto de resultados retornados.
<i>vaRótulosColunas</i>	Um valor booleano opcional que especifica se o banco de dados deve retornar ou não rótulos de colunas para as colunas em cada conjunto de resultados retornados.
<i>vaComentáriosColunas</i>	Um valor booleano opcional que especifica se o banco de dados deve retornar ou não comentários de colunas para as colunas em cada conjunto de resultados retornados.

### Valor de Retorno

Zero se tiver êxito, diferente de zero se não tiver êxito. Se o valor de retorno não for zero, você pode chamar `GetLastErrorString()` ou `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` ou `GetLastSQLState()` para obter informações de erro adicionais.

### ExecuteStored ProcedureEx()

```
short ExecuteStoredProcedureEx(long IDConsulta, VARIANT*  
Resultado, [VARIANT vaConsolidarOK], [VARIANT vaMáxConjResultados],  
[VARIANT vaNomesColunas], [VARIANT vaRótulosColunas], [VARIANT  
vaComentáriosColunas])
```

### Descrição

Esta função executa uma instrução SQL que utiliza o verbo `CALL` do SQL para executar um procedimento armazenado no servidor do banco de dados. Utilize `ExecuteStoredProcedureEx()` quando o procedimento armazenado retornar resultados (ao invés de, ou além de, consultas de resultados). Para procedimentos armazenados que retornam resultados, utilize `ExecuteStoredProcedureEx()`.

Para inicializar um procedimento armazenado para execução com `ExecuteStoredProcedure()`, chame primeiro `InitializeQuery()` especificando uma instrução SQL que utilize a instrução `CALL`. O nome do procedimento armazenado deve ser especificado como um literal na instrução `CALL`. Quaisquer parâmetros especificados na instrução `CALL` (constantes ou não) são ignorados. Ao invés disso, utilize `AddHostVariable()` para especificar as variáveis de entrada e saída.

Se o procedimento armazenado retornar conjuntos de resultados, chame `GetStoredProcedureResultSets()` para recuperar os IDs de consulta para os conjuntos de resultados.

## Parâmetros

Nome	Descrição
<i>IDConsulta</i>	O ID da consulta, conforme retornado de <code>InitializeQuery()</code> . O texto SQL para a consulta deve especificar uma instrução <code>CALL</code> .
<i>Resultado</i>	Um ponteiro para uma <code>VARIANT</code> na qual o resultado é armazenado. O resultado é uma matriz (tipo de variante <code>VT_ARRAY</code>   <code>VT_VARIANT</code> ) que contém um valor para cada coluna no resultado.  Cada valor é especificado em seu tipo de dados nativo ou no tipo de dados da variante mais próximo. Os tipos de retorno suportados são: <code>string</code> (tipo de variante <code>VT_BSTR</code> ), <code>float</code> (tipo de variante <code>VT_R4</code> ), <code>double</code> (tipo de variante <code>VT_R8</code> ), <code>short</code> (tipo de variante <code>VT_I2</code> ), <code>long</code> (tipo de variante <code>VT_I4</code> ) e <code>binary</code> (tipo de variante <code>VT_UI1</code>   <code>VT_ARRAY</code> ).  Você deve inicializar adequadamente a <code>VARIANT</code> antes de chamar esta função. O Visual Basic faz isto automaticamente. Programadores do Visual C++ devem chamar <code>VariantInit()</code> .
<i>vaConsolidarOK</i>	Um valor booleano opcional que especifica se o procedimento armazenado pode consolidar a unidade de trabalho ou se esta operação deve ser restrita. O valor padrão é <code>True</code> .
<i>vaMáxConjResultados</i>	Um valor numérico opcional que especifica o número máximo de conjuntos de resultados que o procedimento armazenado deve ter permissão para retornar. Especifique zero se não desejar que o procedimento armazenado retorne conjuntos de resultados ou se o servidor do banco de dados não suportar o retorno de conjuntos de resultados de procedimentos armazenados sobre <code>DRDA</code> .
<i>vaNomesColunas</i>	Um valor booleano opcional que especifica se o banco de dados deve retornar ou não nomes de colunas para as colunas em cada conjunto de resultados retornados.

<i>vaRótulosColunas</i>	Um valor booleano opcional que especifica se o banco de dados deve retornar ou não rótulos de colunas para as colunas em cada conjunto de resultados retornados.
<i>vaComentáriosColunas</i>	Um valor booleano opcional que especifica se o banco de dados deve retornar ou não comentários de colunas para as colunas em cada conjunto de resultados retornados.

### Valor de Retorno

Zero se tiver êxito, diferente de zero se não tiver êxito. Se o valor de retorno não for zero, você pode chamar `GetLastErrorString()` ou `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` ou `GetLastSQLState()` para obter informações de erro adicionais.

### Export()

`short Export(long IDConsulta, long PrimeiraLinha, long PrimeiraCol, long ÚltimaLinha, long ÚltimaCol, short Formato, short DelimitadorCadeia, short DelimitadorColuna, BOOL IncluirTítulosCol, BSTR NomeArquivo, [VARIANT FormatoDataHora])`

### Descrição

Esta função copia o intervalo de linhas e colunas especificado para a Área de transferência. Se você não tiver recuperado dados de linhas para todas as linhas que deseja copiar para a Área de transferência, chame `CompleteQuery()` antes de chamar esta função. Uma mensagem de erro será retornada se você tentar copiar linhas que não foram recuperadas do banco de dados.

### Parâmetros

Nome	Descrição
<i>IDConsulta</i>	O ID da consulta, conforme retornado de <code>InitializeQuery()</code> .
<i>PrimeiraLinha</i>	A primeira linha que você deseja incluir na exportação.
<i>PrimeiraCol</i>	A primeira coluna que você deseja incluir na exportação.
<i>ÚltimaLinha</i>	A última linha que você deseja incluir na cópia ou -1 se todas as linhas forem incluídas.
<i>ÚltimaCol</i>	A última coluna que você deseja incluir na cópia ou -1 se todas as colunas forem incluídas.
<i>IncluirTítulosCol</i>	Utilize um valor diferente de zero para incluir os títulos das colunas na primeira linha e zero para não inclui-los.
<i>NomeArquivo</i>	Uma cadeia que contém o nome do arquivo no qual você deseja gravar a exportação.

<i>FormatoDataHora</i>	Opcionalmente, o formato a utilizar para valores de data e hora. Os valores válidos são 0 (formato ISO), 1 (formato EUA), 2 (formato EUR), 3 (formato JIS) ou 4 (formato do Painel de controle do Windows). O valor padrão é 4.
------------------------	---

**Nota:** O valor da primeira linha em um conjunto de resultados é 0 e o valor da última linha é um a menos que o número total de linhas. O valor da primeira coluna em um conjunto de resultados é 0 e o valor da última coluna é um a menos que o número total de colunas.

Nome	Descrição
<i>Formato</i>	Especifica o formato da saída.

Valor	Significado
0 (RSEF_TEXT)	O arquivo de saída é gravado em formato de texto corrido.
1 (RSEF_HTML)	O arquivo de saída é gravado em formato HTML e os dados são organizados em uma tabela HTML.
2 (RSEF_CSV)	O arquivo de saída é gravado em formato CSV (valores separados por vírgula).
3 (RSEF_PCIXF)	O arquivo de saída é gravado em formato PC/IXF.
4 (RSEF_S370IXF)	O arquivo de saída é gravado no formato IXF do System/370.

Nome	Descrição
<i>DelimitadorCadeia</i>	Especifica o delimitador de cadeia. Este parâmetro é ignorado se <i>Formato</i> for RSEF_HTML.

Valor	Significado
0 (RSSD_NONE)	Não é utilizado delimitador de cadeia.
1 (RSSD_SINGLEQUOTE)	As cadeias são delimitadas por um caractere de aspas simples (').
2 (RSSD_DOUBLEQUOTE)	As cadeias são delimitadas por um caractere de aspas duplas (").

Nome	Descrição
<i>DelimitadorColuna</i>	Especifica o delimitador de coluna. Este parâmetro é ignorado se <i>Formato</i> for RSEF_HTML.

Valor	Significado
0 (RSCD_SPACE)	As colunas são delimitadas por um caractere de espaço ( ).
1 (RSCD_TAB)	As colunas são delimitadas por um caractere de tabulação (\t).
2 (RSCD_COMMA)	As colunas são delimitadas por um caractere de vírgula (,).

### Valor de Retorno

Zero se tiver êxito, diferente de zero se não tiver êxito. Se o valor de retorno for diferente de zero, você pode chamar `GetLastErrorString()` ou `GetLastErrorType()` para obter informações de erro adicionais. Se o conjunto de resultados for vazio ou linhas não tiverem sido recuperadas do banco de dados, um valor diferente de zero é retornado exceto se *PrimeiraLinha*=0 e *ÚltimaLinha*=1. Neste caso, zero é retornado e um arquivo vazio é gravado.

### Tópicos Relacionados

`CopyToClipboard()`

## ExportForm()

short `ExportForm(BSTR ProprietarioNome, BSTR NomeArquivo)`

### Descrição

Esta função exporta o formulário do QMF especificado para o arquivo especificado.

### Parâmetros

Nome	Descrição
<i>ProprietarioNome</i>	Uma cadeia que contém o proprietário e o nome, separados por um ponto, do formulário que você deseja exportar. Por exemplo, Paulo.Consulta2
<i>NomeArquivo</i>	Uma cadeia que contém o nome do arquivo no qual você deseja gravar o formulário exportado.

### Valor de Retorno

Zero se tiver êxito, diferente de zero se não tiver êxito. Se o valor de retorno não for zero, você pode chamar `GetLastErrorString()` ou `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` ou `GetLastSQLState()` para obter informações de erro adicionais.

### Tópicos Relacionados

`PrintReport()`

## ExportReport()

short ExportReport(long *IDConsulta*, short *TipoOrigem*, BSTR *Origem*, BSTR *NomeArquivoSaída*, short *ComprimentoPágina*, short *LarguraPágina*, BOOL *IncluirDataHora*, BOOL *IncluirNúmPágina*, [VARIANT *Formato*], [VARIANT *UsarConfPágForm*])

### Descrição

Esta função cria um relatório para a consulta especificada e o grava em arquivo. Especifique a formatação e layout para o relatório em um formulário do QMF. O arquivo de saída é um arquivo de texto ASCII com cada linha separada por um par de caracteres de retorno de carro e avanço de linha, e cada página separada por um caractere de avanço do papel. É melhor exibir o arquivo de saída utilizando uma fonte de largura fixa.

### Parâmetros

Nome	Descrição
<i>IDConsulta</i>	O ID da consulta, conforme retornado de InitializeQuery().
<i>Origem</i>	O nome (um nome de arquivo ou Proprietário.Nome) do formulário que você deseja utilizar.
<i>NomeArquivoSaída</i>	O nome do arquivo no qual você deseja gravar o relatório.
<i>ComprimentoPágina</i>	O número de linhas em cada página do relatório. Um <i>ComprimentoPágina</i> com valor -1 especifica saída contínua (sem quebras de página a menos que o relatório seja mais largo que <i>LarguraPágina</i> ).
<i>IncluirDataHora</i>	Um valor diferente de zero especifica que a data e a hora são incluídas no final de cada página. Zero especifica que a data e a hora não são incluídos.
<i>IncluirNúmPágina</i>	Um valor diferente de zero especifica que os números de página são incluídos no final de cada página. Zero especifica que os números de página não são incluídos.
<i>Formato</i>	Opcionalmente, especifica o formato do relatório exportado. Se for zero, o formato é texto corrido, especificando que a saída deve ser exatamente a produzida pelo formulário (texto ou HTML, dependendo do tipo de formulário). Se for um valor diferente de zero, o formato é HTML, especificando que a saída deve ser HTML. Com formulários não-HTML, a saída é convertida para HTML incluindo-se tags HTML no início e no final da saída. O valor padrão é zero.
<i>FormatoDataHora</i>	Opcionalmente, o formato a utilizar para valores de data e hora. Os valores válidos são 0 (formato ISO), 1 (formato EUA), 2 (formato EUR), 3 (formato JIS) ou 4 (formato do Painel de controle do Windows). O valor padrão é 4.
<i>Formato</i>	O formato do arquivo de saída.

UsarConfPágForm	Opcionalmente, se for um valor diferente de zero, especifica que os parâmetros <i>ComprimentoPágina</i> , <i>LarguraPágina</i> , <i>IncluirDataHora</i> e <i>IncluirNúmPágina</i> devem ser ignorados e os valores para eles devem ser tomados dos valores salvos com o formulário especificado. O valor padrão é zero.
-----------------	---

Valor	Significado
0 (RSF_DEFAULT)	Utiliza o formulário padrão. <i>NomeFormulário</i> deve ser uma cadeia vazia.
1 (RSF_DATABASE)	Utiliza um formulário do banco de dados. Especifique o proprietário e nome do formulário (Proprietário.Nome) no parâmetro <i>NomeFormulário</i> . Para utilizar um formulário localizado em um servidor de banco de dados diferente, primeiro utilize <i>ExportForm()</i> para exportar o formulário para um arquivo e, em seguida, especifique <i>TipoOrigem</i> como <i>RSF_FILE</i> .
2 (RSF_FILE)	Utiliza um formulário contido em um arquivo. Especifique o nome do arquivo no parâmetro <i>NomeFormulário</i> .

### Valor de Retorno

Zero se tiver êxito, diferente de zero se não tiver êxito. Se o valor de retorno não for zero, você pode chamar *GetLastErrorString()* ou *GetLastErrorType()*, *GetLastSQLCode()*, *GetLastSQLError()* ou *GetLastSQLState()* para obter informações de erro adicionais.

### Tópicos Relacionados

*ExportForm()*

### FastSaveData()

*short FastSaveData(long IDConsulta, BOOL Substituir, BSTR NomeTabela, BSTR NomeEspaçoTabela, [VARIANT Comentário])*

### Descrição

Esta função cria um relatório para a consulta especificada e o grava em arquivo. Especifique a formatação e layout para o relatório em um formulário do QMF. O arquivo de saída é um arquivo de texto ASCII com cada linha separada por um par de caracteres de retorno de carro e avanço de linha, e cada página separada por um caractere de avanço do papel. É melhor exibir o arquivo de saída utilizando uma fonte de largura fixa.

### Parâmetros

Nome	Descrição
------	-----------

<i>IDConsulta</i>	O ID da consulta, conforme retornado de <code>InitializeQuery()</code> .
<i>Substituir</i>	Utilize um valor diferente de zero se desejar que os dados especificados substituam quaisquer dados existentes na tabela. Utilize zero se desejar que os dados especificados sejam anexados a quaisquer dados existentes na tabela.
<i>NomeTabela</i>	O nome da tabela na qual você deseja armazenar os dados. Se a tabela não existir, o QMF para Windows a criará.
<i>NomeEspaçoTabela</i>	O nome do espaço da tabela no qual a tabela existe ou é criada. Se <i>NomeEspaçoTabela</i> for NULL ou uma cadeia vazia, o QMF para Windows utilizará o espaço de tabela padrão. Se você tiver configurado o QMF para Windows para sempre utilizar o espaço de tabela padrão, este parâmetro será ignorado. Consulte <code>RSR_SDDIFFERENTTS</code> na descrição de <code>GetResourceLimit()</code> .
<i>Comentário</i>	Opcionalmente, uma cadeia que especifica um comentário para a tabela na qual os dados são salvos.

### Valor de Retorno

Zero se tiver êxito, diferente de zero se não tiver êxito. Se o valor de retorno não for zero, você pode chamar `GetLastErrorString()` ou `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` ou `GetLastSQLState()` para obter informações de erro adicionais.

### FetchNextRow()

`short FetchNextRow(long IDConsulta, VARIANT* Linha)`

### Descrição

Esta função busca a próxima linha de dados no banco de dados.

### Parâmetros

Nome	Descrição
<i>IDConsulta</i>	O ID da consulta, conforme retornado de <code>InitializeQuery()</code> .

---

Linha	<p>Um ponteiro para uma VARIANT na qual o resultado é armazenado. O resultado é uma matriz (tipo de variante VT_ARRAY   VT_VARIANT) que contém um valor para cada coluna no resultado. Chame GetColumnCount() para determinar o número de valores na matriz.</p> <p>Cada valor é especificado em seu tipo de dados nativo ou no tipo de dados da variante mais próximo. Os tipos de retorno suportados são: string (tipo de variante VT_BSTR), float (tipo de variante VT_R4), double (tipo de variante VT_R8), short (tipo de variante VT_I2), long (tipo de variante VT_I4) e binary (tipo de variante VT_UI1   VT_ARRAY).</p> <p>Quando o final do conjunto de resultados tiver sido alcançado (não há mais linhas a serem buscadas) ou se o conjunto de resultados for vazio, o resultado será vazio (tipo de variante VT_EMPTY), ao invés de uma matriz .</p> <p>Você deve inicializar adequadamente a VARIANT antes de chamar esta função. O Visual Basic faz isto automaticamente. Programadores do Visual C++ devem chamar VariantInit().</p>
-------	---

---

**Nota:** Devido a um erro no Microsoft Excel 7.0 e no Microsoft Access 7.0 (e possivelmente em outros produtos de 32 bits da Microsoft que utilizam o Visual Basic for Applications), os dados de cadeia nas variáveis Variante recebidos do QMF para Windows podem não ser convertidos do Unicode (utilizado pelo OLE) para ANSI (utilizado pelo VBA). Quando isto ocorre, somente o primeiro caractere da cadeia é exibido. Para corrigir este problema, defina a variável como uma cadeia vazia antes de chamar a função do QMF para Windows que utiliza a variável.

### Valor de Retorno

Zero se tiver êxito, diferente de zero se não tiver êxito. Quando o final do conjunto de resultados é alcançado, o valor de retorno é -1. Se o valor de retorno não for zero, você pode chamar GetLastErrorString() ou GetLastErrorType(), GetLastErrorSQLCode(), GetLastErrorSQLError() ou GetLastErrorSQLState() para obter informações de erro adicionais.

### Tópicos Relacionados

FetchNextRows()

### FetchNextRowEx()

short FetchNextRowEx(long *IDConsulta*)

### Descrição

Esta função busca a próxima linha de dados no banco de dados. Você pode utilizar esta função em ambientes que não suportam matrizes VARIANT,

como o Microsoft Access 2.0. Utilize esta função em conjunto com `GetColumnValue()` para recuperar os dados em cada coluna para a linha atual.

### Parâmetros

Nome	Descrição
<i>IDConsulta</i>	O ID da consulta, conforme retornado de <code>InitializeQuery()</code> .

### Valor de Retorno

Zero se tiver êxito, diferente de zero se não tiver êxito. Quando o final do conjunto de resultados é alcançado, o valor de retorno é -1. Se o valor de retorno não for zero, você pode chamar `GetLastErrorString()` ou `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` ou `GetLastSQLState()` para obter informações de erro adicionais.

### Tópicos Relacionados

`FetchNextRowsEx()`

### FetchNextRows()

`short FetchNextRows(long IDConsulta, VARIANT* Linhas, long* NúmLinhas)`

### Descrição

Esta função busca as próximas *NúmLinhas* de dados do banco de dados.

### Parâmetros

Nome	Descrição
<i>IDConsulta</i>	O ID da consulta, conforme retornado de <code>InitializeQuery()</code> .

Linha	<p>Um ponteiro para uma VARIANT na qual o resultado é armazenado. O resultado é uma matriz bidimensional (tipo de variante VT_ARRAY   VT_VARIANT) que contém um valor para cada coluna em cada linha. Chame GetColumnCount() para determinar o número de colunas na matriz. As dimensões da matriz são [NúmLinhas][ContColunas], mesmo se o número de linhas não buscadas no conjunto de resultados for menor que NúmLinhas (neste caso, a matriz contém entradas adicionais, não utilizadas).</p> <p>Cada valor é especificado em seu tipo de dados nativo ou no tipo de dados da variante mais próximo. Os tipos de retorno suportados são: string (tipo de variante VT_BSTR), float (tipo de variante VT_R4), double (tipo de variante VT_R8), short (tipo de variante VT_I2), long (tipo de variante VT_I4) e binary (tipo de variante VT_UI1   VT_ARRAY).</p> <p>Quando o final do conjunto de resultados tiver sido alcançado (não há mais linhas a serem buscadas) ou se o conjunto de resultados for vazio, o resultado será vazio (tipo de variante VT_EMPTY), ao invés de uma matriz</p> <p>Você deve inicializar adequadamente a VARIANT antes de chamar esta função. O Visual Basic faz isto automaticamente. Programadores do Visual C++ devem chamar VariantInit().</p>
NúmLinhas	<p>Um ponteiro para um inteiro longo que contém o número de linhas a serem buscadas. Se o número de linhas não buscadas no conjunto de resultados for menor que NúmLinhas, NúmLinhas é redefinido para o número real de linhas contidas no resultado.</p>

**Nota:** Devido a um erro no Microsoft Excel 7.0 e no Microsoft Access 7.0 (e possivelmente em outros produtos de 32 bits da Microsoft que utilizam o Visual Basic for Applications), os dados de cadeia nas variáveis Variante recebidos do QMF para Windows podem não ser convertidos de Unicode (utilizado pelo OLE) para ANSI (utilizado pelo VBA). Quando isto ocorre, somente o primeiro caractere da cadeia é exibido. Para corrigir este problema, defina a variável como uma cadeia vazia antes de chamar a função do QMF para Windows que utiliza a variável.

### Valor de Retorno

Zero se tiver êxito, diferente de zero se não tiver êxito. Quando o final do conjunto de resultados é alcançado, o valor de retorno é -1. Se o valor de retorno não for zero, você pode chamar GetLastErrorString() ou GetLastErrorType(), GetLastErrorSQLCode(), GetLastErrorSQLError() ou GetLastErrorSQLState() para obter informações de erro adicionais.

## Tópicos Relacionados

FetchNextRow()

### FetchNextRowsEx()

short FetchNextRowsEx(long *IDConsulta*, long\* *NúmLinhas*)

#### Descrição

Esta função busca as próximas *NúmLinhas* de dados do banco de dados. Você pode utilizar esta função em ambientes que não suportam matrizes VARIANT, como o Microsoft Access 2.0. Utilize esta função em conjunto com GetColumnValueEx() para recuperar os dados em cada coluna para uma determinada linha.

#### Parâmetros

Nome	Descrição
<i>IDConsulta</i>	O ID da consulta, conforme retornado de InitializeQuery().
<i>NúmLinhas</i>	Um ponteiro para um inteiro longo que contém o número de linhas a serem buscadas. Se o número de linhas não buscadas no conjunto de resultados for menor que <i>NúmLinhas</i> , <i>NúmLinhas</i> é redefinido para o número real de linhas contidas no resultado.

#### Valor de Retorno

Zero se tiver êxito, diferente de zero se não tiver êxito. Quando o final do conjunto de resultados é alcançado, o valor de retorno é -1. Se o valor de retorno não for zero, você pode chamar GetLastErrorString() ou GetLastErrorType(), GetLastSQLCode(), GetLastSQLError() ou GetLastSQLState() para obter informações de erro adicionais.

## Tópicos Relacionados

FetchNextRowEx()

### FlushQMFCache()

void FlushQMFCache()

#### Descrição

Esta função instrui o QMF para Windows para que descarregue seu cache de informações do QMF, descartando seu conteúdo. Na próxima vez que o QMF para Windows precisar de informações de QMF, ele o obterá do banco de dados. Normalmente, o QMF para Windows armazena em cache informações de QMF obtidas do banco de dados para reduzir o tráfego neste e para melhorar o desempenho. Chame esta função antes de chamar GetQMFObjectInfo(), GetQMFObjQueryText() ou GetQMFObjList() para assegurar que as informações retornadas estejam atualizadas.

### Valor de Retorno

Nenhum.

## GetColumnCount()

long GetColumnCount(long *IDConsulta*)

### Descrição

Esta função retorna o número de colunas no conjunto de resultados.

### Parâmetros

Nome	Descrição
<i>IDConsulta</i>	O ID da consulta, conforme retornado de InitializeQuery().

### Valor de Retorno

O número de colunas em cada linha, se tiver êxito. Se não tiver êxito, 0 ou -1. Se o valor de retorno for 0 ou -1, você pode chamar GetLastErrorString() ou GetLastErrorType(), GetLastErrorSQLCode(), GetLastErrorSQLError() ou GetLastErrorSQLState() para obter informações de erro adicionais.

## GetColumnDataValue()

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

### Descrição

Esta função retorna o valor de dados para a coluna especificada em *Índice* para a linha de dados atual. Após a chamada desta função, a propriedade *Value* pode ser interrogada para se obter o valor retornado. Utilize esta função com FetchNextRowEx() para acessar os dados em uma única linha de dados.

### Parâmetros

Nome	Descrição
<i>IDConsulta</i>	O ID da consulta, conforme retornado de InitializeQuery().
<i>Índice</i>	O índice baseado em zero da linha de dados a ser recuperada.

### Valor de Retorno

Zero se tiver êxito, diferente de zero se não tiver êxito. Se o valor de retorno não for zero, você pode chamar GetLastErrorString() ou GetLastErrorType(), GetLastErrorSQLCode(), GetLastErrorSQLError() ou GetLastErrorSQLState() para obter informações de erro adicionais.

## GetColumnHeader()

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

### Descrição

Esta função retorna o cabeçalho da coluna (nome da coluna) associado ao índice *Índice*.

### Parâmetros

Nome	Descrição
<i>IDConsulta</i>	O ID da consulta, conforme retornado de <code>InitializeQuery()</code> .
Índice	O índice baseado em zero da linha de dados a ser recuperada.
Resultado	Zero se tiver êxito, diferente de zero se não tiver êxito. Se o Resultado for diferente de zero, você pode chamar <code>GetLastErrorString()</code> ou <code>GetLastErrorType()</code> , <code>GetLastSQLCode()</code> , <code>GetLastSQLError()</code> ou <code>GetLastSQLState()</code> para obter informações de erro adicionais.

**Nota:** Os títulos de colunas não estão disponíveis para instruções SQL estáticas. Para IDs de consulta retornados de `InitializeStaticQuery()`, `GetColumnHeader` retorna uma cadeia do formulário "Coln" em que "n" é o número da coluna.

### Valor de Retorno

A cadeia retornada representa o nome da coluna, conforme especificado no parâmetro *Índice*.

## GetColumnHeaderEx()

short `GetColumnHeaderEx(long IDConsulta, long Índice)`

### Descrição

Esta função retorna o cabeçalho da coluna (nome da coluna) associado ao índice *Índice*. Após a chamada desta função, a propriedade *Value* pode ser interrogada para se obter o valor retornado.

### Parâmetros

Nome	Descrição
<i>IDConsulta</i>	O ID da consulta, conforme retornado de <code>InitializeQuery()</code> .
Índice	O índice baseado em zero da linha de dados a ser recuperada.

**Nota:** Os títulos de colunas não estão disponíveis para instruções SQL estáticas. Para IDs de consulta retornados de `InitializeStaticQuery()`, `GetColumnHeader` retorna uma cadeia do formulário "Coln" em que "n" é o número da coluna.

### Valor de Retorno

Zero se tiver êxito, diferente de zero se não tiver êxito. Se o valor de retorno for zero, consulte a propriedade *Value* para a cadeia que representa o nome da coluna. Se o valor de retorno não for zero, você pode chamar `GetLastErrorString()` ou `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` ou `GetLastSQLState()` para obter informações de erro adicionais.

## GetColumnHeadings()

short GetColumnHeadings(long *IDConsulta*, VARIANT\* *Títulos*)

### Descrição

Esta função retorna os títulos de colunas (também referidos como nomes de colunas).

### Parâmetros

Nome	Descrição
<i>IDConsulta</i>	O ID da consulta, conforme retornado de <code>InitializeQuery()</code> .
Títulos	Um ponteiro para uma VARIANT na qual o resultado é armazenado. O resultado é uma matriz de cadeias (tipo de variante VT_ARRAY   VT_BSTR) que contém uma cadeia para cada título de coluna.  Você deve inicializar adequadamente a VARIANT antes de chamar esta função. O Visual Basic faz isto automaticamente. Programadores do Visual C++ devem chamar <code>VariantInit()</code> .

**Nota:** Devido a um erro no Microsoft Excel 7.0 e no Microsoft Access 7.0 (e possivelmente em outros produtos de 32 bits da Microsoft que utilizam o Visual Basic for Applications), os dados de cadeia nas variáveis Variante recebidos do QMF para Windows podem não ser convertidos de Unicode (utilizado pelo OLE) para ANSI (utilizado pelo VBA). Quando isto ocorre, somente o primeiro caractere da cadeia é exibido. Para corrigir este problema, defina a variável como uma cadeia vazia antes de chamar a função do QMF para Windows que utiliza a variável.

**Nota:** Os títulos de colunas não estão disponíveis para instruções SQL estáticas. Para IDs de consultas retornados de `InitializeStaticQuery()`, `GetColumnHeadings` retorna as cadeias "Col1", "Col2", etc.

### Valor de Retorno

Zero se tiver êxito, diferente de zero se não tiver êxito. Se o valor de retorno não for zero, você pode chamar `GetLastErrorString()` ou `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` ou `GetLastSQLState()` para obter informações de erro adicionais.

## GetColumnValue()

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

### Descrição

Esta função retorna o valor de dados para a coluna especificada em *Índice* para a linha de dados atual. Utilize esta função com `FetchNextRowEx()` para acessar os dados em uma única linha de dados.

### Parâmetros

Nome	Descrição
<i>IDConsulta</i>	O ID da consulta, conforme retornado de <code>InitializeQuery()</code> .
Índice	O índice baseado em zero da linha de dados a ser recuperada.
Valor	Um ponteiro para uma VARIANT na qual você deseja armazenar os resultados. O resultado é um valor de dados baseado no tipo de variante.  Você deve inicializar adequadamente a VARIANT antes de chamar esta função. O Visual Basic faz isto automaticamente. Programadores do Visual C++ devem chamar <code>VariantInit()</code> .

### Valor de Retorno

Zero se tiver êxito, diferente de zero se não tiver êxito. Se o valor de retorno não for zero, você pode chamar `GetLastErrorString()` ou `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` ou `GetLastSQLState()` para obter informações de erro adicionais.

## GetColumnValueEx()

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

### Descrição

Esta função retorna o valor de dados para a coluna especificada em *ÍndiceCol* para a linha de dados especificados em *ÍndiceLinha*. Utilize esta função com `FetchNextRowsEx()` para acessar os dados em uma única linha de dados.

### Parâmetros

Nome	Descrição
<i>IDConsulta</i>	O ID da consulta, conforme retornado de <code>InitializeQuery()</code> .

ÍndiceLinha	O índice baseado em zero da linha a ser recuperada.
ÍndiceCol	O índice baseado em zero da coluna a ser recuperada.
Valor	Um ponteiro para uma VARIANT na qual você deseja armazenar o resultado. Você pode consultar a variante resultante para descobrir o tipo de dados para processamento adicional.  Você deve inicializar adequadamente a VARIANT antes de chamar esta função. O Visual Basic faz isto automaticamente. Programadores do Visual C++ devem chamar VariantInit().

### Valor de Retorno

Zero se tiver êxito, diferente de zero se não tiver êxito. Se o valor de retorno não for zero, você pode chamar GetLastErrorString() ou GetLastErrorType(), GetLastErrorSQLCode(), GetLastErrorSQLError() ou GetLastErrorSQLState() para obter informações de erro adicionais.

### GetDefaultServerName()

BSTR GetDefaultServerName()

#### Descrição

Esta função retorna uma cadeia que contém o nome do servidor padrão.

#### Valor de Retorno

Uma cadeia que especifica o nome do servidor padrão.

### GetGlobalVariable()

BSTR GetGlobalVariable(BSTR *Nome*)

#### Descrição

Esta função recupera o valor da variável global especificada.

#### Parâmetros

Nome	Descrição
Nome	Uma cadeia que contém o nome da variável que você deseja definir.

#### Valor de Retorno

Uma cadeia que contém o valor da variável global, ou NULL se a variável não tiver valor ou ocorrer um erro.

### GetHostVariableNames()

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

### Descrição

Esta função retorna uma matriz dos nomes de todas as variáveis de host referenciadas na consulta especificada. A consulta deve ser uma consulta estática referenciando variáveis de host (ou armazenadas com a consulta do QMF, ou criadas por `AddHostVariable()`).

### Parâmetros

Nome	Descrição
<i>IDConsulta</i>	O ID da consulta, conforme retornado de <code>InitializeQuery()</code> .
Nomes	Um ponteiro para uma VARIANT na qual você deseja armazenar a matriz de resultados.

### Valor de Retorno

Zero se tiver êxito, diferente de zero se não tiver êxito. Se o valor de retorno for diferente de zero, você pode chamar `GetLastErrorString()` para obter informações de erro adicionais.

## GetHostVariableTypes()

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

### Descrição

Esta função retorna uma matriz de tipos de dados de todas as variáveis de host referenciadas na consulta especificada. A consulta deve ser uma consulta estática referenciando variáveis de host (ou armazenadas com a consulta do QMF, ou criadas por `AddHostVariable()`). Consulte `AddHostVariable()` para obter uma lista dos tipos de dados que podem ser retornados.

### Parâmetros

Nome	Descrição
<i>IDConsulta</i>	O ID da consulta, conforme retornado de <code>InitializeQuery()</code> .
Tipos	Um ponteiro para uma VARIANT na qual você deseja armazenar a matriz de resultados.

### Valor de Retorno

Zero se tiver êxito, diferente de zero se não tiver êxito. Se o valor de retorno for diferente de zero, você pode chamar `GetLastErrorString()` para obter informações de erro adicionais.

## GetLastErrorString()

BSTR `GetLastErrorString()`

### Descrição

Esta função retorna uma cadeia que contém informações sobre o erro mais recente. Se você chamar esta função após uma função que foi executada com êxito (sem erros), então esta função retornará informações sobre o último erro que ocorreu durante uma chamada de função anterior. Para evitar confusão, chame sempre esta função imediatamente após chamar uma função que retornou um erro.

### Valor de Retorno

Uma cadeia que contém informações de erro. Se não ocorreram erros desde que você criou o objeto da API do QMF, será retornado NULL.

### Tópicos Relacionados

GetLastErrorType()

GetLastSQLCode()

GetLastSQLError()

GetLastSQLState()

## GetLastErrorType()

short GetLastErrorType()

### Descrição

Esta função retorna o tipo do erro mais recente. Se você chamar esta função após uma função que foi executada com êxito (sem erros), então esta função retornará informações sobre o último erro que ocorreu durante uma chamada de função anterior. Para evitar confusão, chame sempre esta função imediatamente após chamar uma função que retornou um erro.

### Valor de Retorno

O valor retornado indica o tipo de erro:

Valor	Significado
0 (RS_ERROR_NONE)	Não ocorreram erros desde que o objeto da API do QMF para Windows foi criado.
1 (RS_ERROR_SQL)	Ocorreu um erro de SQL. Se o erro ocorreu durante uma chamada a uma função que toma <i>IDConsulta</i> como argumento, chame <i>Close()</i> para fechar a consulta. Nenhuma reversão é executada. Você pode continuar a utilizar o objeto da API do QMF para Windows, embora possa encontrar erros adicionais.

2 (RS_ERROR_USER_CANCEL)	Um usuário cancelou a operação, geralmente clicando em Cancelar na janela ocupada. Isto faz com que o QMF para Windows realize uma reversão implícita (invalidando todos os IDs de consulta pendentes) e destrua a conexão com o banco de dados. Você deve chamar InitializeServer() ou ReinitializeServer() se quiser continuar.
3 (RS_ERROR_FATAL_GOV)	Ocorreu um erro fatal do controlador. Uma possibilidade é que o tempo limite da API do QMF para Windows expirou porque o máximo tempo inativo permissível foi excedido. Isto faz com que o QMF para Windows realize uma reversão implícita (invalidando todos os IDs de consulta pendentes) e destrua a conexão com o banco de dados. Você deve chamar InitializeServer() ou ReinitializeServer() se quiser continuar.
4 (RS_ERROR_NONFATAL_GOV)	Ocorreu um erro não fatal do controlador. Ou o número máximo de colunas a buscar permissível foi excedido, ou o verbo SQL não é permitido. Se o erro ocorreu durante uma chamada a uma função que toma <i>IDConsulta</i> como argumento, chame Close() para fechar essa consulta. Nenhuma reversão é realizada e a conexão com o banco de dados não é afetada, portanto você pode continuar a utilizar o objeto da API do QMF para Windows.
5 (RS_ERROR_OTHER)	Ocorreu um erro geral. Nenhuma reversão é executada. Você pode continuar a utilizar o objeto da API do QMF para Windows, embora possa encontrar erros adicionais.

### Tópicos Relacionados

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

### GetLastSQLCode()

long GetLastSQLCode()

### Descrição

Esta função retorna o código SQL do erro mais recente. Se você chamar esta função após uma função que foi executada com êxito (sem erros), então esta função retornará informações sobre o último erro que ocorreu durante uma

chamada de função anterior. Para evitar confusão, chame sempre esta função imediatamente após chamar uma função que retornou um erro.

### Valor de Retorno

Os códigos SQL do erro mais recente. Se não ocorreram erros desde que você criou o objeto da API do QMF para Windows ou o erro mais recente não foi um erro de SQL, zero é retornado.

### Tópicos Relacionados

GetLastErrorString()

GetLastErrorType()

GetLastSQLError()

GetLastSQLState()

## GetLastSQLError()

VARIANT GetLastError()

### Descrição

Esta função retorna informações de erro de SQL detalhadas para o erro mais recente. Se você chamar esta função após uma função que foi executada com êxito (sem erros), então esta função retornará informações sobre o último erro que ocorreu durante uma chamada de função anterior. Para evitar confusão, chame sempre esta função imediatamente após chamar uma função que retornou um erro.

### Valor de Retorno

Uma matriz (tipo de variante VT\_ARRAY | VT\_VARIANT) que contém informações de erro. Se não ocorreram erros desde que você criou o objeto da API do QMF para Windows ou o erro mais recente não foi um erro de SQL, será retornado vazio (tipo de variante VT\_EMPTY). A matriz tem o seguinte formato:

Elemento	Tipo	Conteúdo
0	long (VT_I4)	Código
1	string (VT_BSTR)	Estado
2	string (VT_BSTR)	ProcErr
3	string (VT_BSTR)	NomeRDB
4	long (VT_I4)	ErrD1
5	long (VT_I4)	ErrD2
6	long (VT_I4)	ErrD3
7	long (VT_I4)	ErrD4
8	long (VT_I4)	ErrD5
9	long (VT_I4)	ErrD6

10	string (VT_BSTR)	Avis0
11	string (VT_BSTR)	Avis1
12	string (VT_BSTR)	Avis2
13	string (VT_BSTR)	Avis3
14	string (VT_BSTR)	Avis4
15	string (VT_BSTR)	Avis5
16	string (VT_BSTR)	Avis6
17	string (VT_BSTR)	Avis7
18	string (VT_BSTR)	Avis8
19	string (VT_BSTR)	Avis9
20	string (VT_BSTR)	AvisA
21	string (VT_BSTR)	TokensMensagem

### Tópicos Relacionados

GetLastErrorString()

GetLastErrorType()

GetLastSQLCode()

GetLastSQLState()

## GetLastSQLState()

BSTR GetLastErrorSQLState()

### Descrição

Esta função retorna o estado SQL do erro mais recente. Se você chamar esta função após uma função que foi executada com êxito (sem erros), então esta função retornará informações sobre o último erro que ocorreu durante uma chamada de função anterior. Para evitar confusão, chame sempre esta função imediatamente após chamar uma função que retornou um erro.

### Valor de Retorno

Uma cadeia contendo o código SQL para o erro mais recente. Se não ocorreram erros desde que você criou o objeto da API do QMF para Windows ou o erro mais recente não foi um erro de SQL, será retornado NULL.

### Tópicos Relacionados

GetLastErrorString()

GetLastErrorType()

GetLastSQLCode()

GetLastSQLError()

## GetOption()

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

## Descrição

Obtém o valor da opção especificado no QMF para Windows.

## Parâmetros

Nome	Descrição
<i>Opção</i>	Especifica a opção a ser recuperada.
Valor	Significado
0 (RSO_SERVER_DEFINITION_FILE)	Nome do arquivo de definição do servidor.
1 (RSO_CPIC_DLL)	Nome do arquivo da DLL do provedor CPI-C.
2 (RSO_CPIC_TIMEOUT_WARNING)	Tempo limite de aviso do CPI-C (em segundos). Este limite não é utilizado pela API do QMF para Windows.
3 (RSO_CPIC_TIMEOUT_CANCEL)	Tempo limite de cancelamento do CPI-C (em segundos).
4 (RSO_TCP_TIMEOUT_WARNING)	Tempo limite de aviso do TCP (em segundos). Este limite não é utilizado pela API do QMF para Windows.
5 (RSO_TCP_TIMEOUT_CANCEL)	Tempo limite de cancelamento do TCP (em segundos).
6 (RSO_DISPLAY_NULLS_STRING)	A cadeia a ser utilizada para exibir valores nulos.
7 (RSO_ENTER_NULLS_STRING)	A cadeia a ser utilizada para inserir valores nulos.
8 (RSO_ENTER_DEFAULTS_STRING)	A cadeia a ser utilizada para inserir valores padrão.
9 (RSO_TRACE_FILE_1)	Nome do arquivo de rastreamento 1.
10 (RSO_TRACE_FILE_2)	Nome do arquivo de rastreamento 2.
11 (RSO_TCP_TRACE_LEVEL)	Nível de rastreamento do TCP.
12 (RSO_CPIC_TRACE_LEVEL)	Nível de rastreamento do CPI-C.
13 (RSO_DDM_TRACE_LEVEL)	Nível de rastreamento do DDM.
Valor	Um ponteiro para uma VARIANT na qual o resultado é armazenado. O resultado é uma matriz (tipo de variante VT_ARRAY   VT_VARIANT) que contém um valor para cada coluna no resultado. Chame GetColumnCount() para determinar o número de valores na matriz. Você deve inicializar adequadamente a VARIANT antes de chamar esta função. O Visual Basic faz isto automaticamente. Programadores do Visual C++ devem chamar VariantInit().

**Nota:** Devido a um erro no Microsoft Excel 7.0 e no Microsoft Access 7.0 (e possivelmente em outros produtos de 32 bits da Microsoft que utilizam o Visual Basic for Applications), os dados de cadeia nas variáveis Variante recebidos do QMF para Windows podem não ser convertidos de Unicode (utilizado pelo OLE) para ANSI (utilizado pelo VBA). Quando isto ocorre, somente o primeiro caractere da cadeia é exibido. Para corrigir este problema, defina a variável como uma cadeia vazia antes de chamar a função do QMF para Windows que utiliza a variável.

### Valor de Retorno

Zero se tiver êxito, diferente de zero se não tiver êxito. Se o valor de retorno for diferente de zero, você pode chamar `GetLastErrorString()` ou `GetLastErrorType()` para obter informações de erro adicionais.

### Tópicos Relacionados

`SetOption()`

## GetOptionEx()

`short GetOptionEx(short Opção)`

### Descrição

Obtém o valor da opção especificado no QMF para Windows. Quando o valor da opção é retornado, você deve consultar a propriedade *Opção* para os dados.

### Parâmetros

Nome	Descrição
Opção	Os valores de opção são os mesmos que os da chamada <code>GetOption()</code> .

### Valor de Retorno

Zero se tiver êxito, diferente de zero se não tiver êxito. Se o valor de retorno for diferente de zero, você pode chamar `GetLastErrorString()` ou `GetLastErrorType()` para obter informações de erro adicionais.

### Tópicos Relacionados

`GetOption()`

`SetOption()`

## GetProcText()

`BSTR GetProcText(long IDProc)`

### Descrição

Esta função retorna o texto que é executado para o procedimento especificado, após a substituição de variáveis. Você deve utilizar SetProcVariable() para definir o valor de quaisquer variáveis utilizadas no procedimento antes de chamar esta função.

### Parâmetros

Nome	Descrição
IDProc	O ID do procedimento, conforme retornado de InitializeProc().

### Valor de Retorno

Se tiver êxito, uma cadeia contendo o texto do procedimento é retornada. Se não tiver êxito, NULL é retornado. Se o valor de retorno for NULL, você pode chamar GetLastErrorString() ou GetLastErrorType() para obter informações de erro adicionais.

## GetProcVariables()

short GetProcVariables(long IDProc, VARIANT\* Variáveis)

### Descrição

Obtém o valor da opção especificado no QMF para Windows.

### Parâmetros

Nome	Descrição
IDProc	O ID do procedimento, conforme retornado de InitializeProc().
Valor	Um ponteiro para uma VARIANT na qual o resultado é armazenado. O resultado é uma matriz de cadeias (tipo de variante VT_ARRAY   VT_BSTR) com cada cadeia contendo o nome de uma variável. Se não existirem variáveis no procedimento, o resultado será vazio (tipo de variante VT_EMPTY). Você deve inicializar adequadamente a VARIANT antes de chamar esta função. O Visual Basic faz isto automaticamente. Programadores do Visual C++ devem chamar VariantInit().

**Nota:** Devido a um erro no Microsoft Excel 7.0 e no Microsoft Access 7.0 (e possivelmente em outros produtos de 32 bits da Microsoft que utilizam o Visual Basic for Applications), os dados de cadeia nas variáveis Variante recebidos do QMF para Windows podem não ser convertidos de Unicode (utilizado pelo OLE) para ANSI (utilizado pelo VBA). Quando isto ocorre, somente o primeiro caractere da cadeia é exibido.

Para corrigir este problema, defina a variável como uma cadeia vazia antes de chamar a função do QMF para Windows que utiliza a variável.

### Valor de Retorno

Zero se tiver êxito, diferente de zero se não tiver êxito. Se não existirem variáveis no procedimento, o resultado será `RS_NO_ERROR_NO_DATA` (-1). Se o valor de retorno for diferente de zero, você pode chamar `GetLastErrorString()` ou `GetLastErrorType()` para obter informações de erro adicionais.

## GetQMFOBJECTINFO()

short GetQMFOBJECTINFO(BSTR ProprietarioeNome, short Tipo, short Hora, VARIANT\* Valor)

### Descrição

Esta função retorna informações sobre um objeto do QMF (ou um formulário, ou uma consulta). As informações retornadas são especificadas pelos parâmetros *Tipo* e *Hora*.

### Parâmetros

Nome	Descrição
ProprietarioeNome	Uma cadeia que contém o proprietário e o nome, separados por um ponto, do objeto cujas informações você deseja recuperar. Por exemplo, Paulo.Consulta2
Valor	Um ponteiro para uma VARIANT na qual o resultado é armazenado. Para <code>RSI_TIMEUSED</code> , <code>RSI_TIMESRUN</code> , <code>RSI_TIMESCANCELLED</code> e <code>RSI_LEVEL</code> , o resultado é um inteiro curto (tipo de variante <code>VT_I2</code> ). Para <code>RSI_RESTRICTED</code> o resultado é um booleano (tipo de variante <code>VT_BOOL</code> ). Para todos os outros, o resultado é uma cadeia (tipo de variante <code>VT_BSTR</code> ). Você deve inicializar adequadamente a VARIANT antes de chamar esta função. O Visual Basic faz isto automaticamente. Programadores do Visual C++ devem chamar <code>VariantInit()</code> .

**Nota:** Devido a um erro no Microsoft Excel 7.0 e no Microsoft Access 7.0 (e possivelmente em outros produtos de 32 bits da Microsoft que utilizam o Visual Basic for Applications), os dados de cadeia nas variáveis Variante recebidos do QMF para Windows podem não ser convertidos de Unicode (utilizado pelo OLE) para ANSI (utilizado pelo VBA). Quando isto ocorre, somente o primeiro caractere da cadeia é exibido. Para corrigir este problema, defina a variável como uma cadeia vazia antes de chamar a função do QMF para Windows que utiliza a

variável.

<i>Tipo</i>	Especifica o tipo de informações a obter.
Valor	Significado
0 (RSI_COMMENT)	Comentário
1 (RSI_LEVEL)	Nível
2 (RSI_TYPE)	Tipo
3 (RSI_SUBTYPE)	Subtipo
4 (RSI_RESTRICTED)	Restrito
5 (RSI_MODEL)	Modelo
6 (RSI_TIMESUSED)	Número de vezes utilizado.
7 (RSI_TIMESRUN)	Número de vezes executado.
8 (RSI_TIMESCANCELLED)	Número de vezes cancelado.
9 (RSI_DATE)	Data da primeira utilização, última utilização ou última modificação.
10 (RSI_TIME)	Hora da primeira utilização, última utilização ou última modificação.
11 (RSI_USERID)	ID do usuário da primeira utilização, última utilização ou última modificação.
12 (RSI_SQLID)	ID SQL da primeira utilização, última utilização ou última modificação.
13 (RSI_ENVIRONMENT)	Ambiente da primeira utilização, última utilização ou última modificação.
14 (RSI_MODE)	Modo da primeira utilização, última utilização ou última modificação.
15 (RSI_COMMAND)	Comando da primeira utilização, última utilização ou última modificação.
<i>Hora</i>	Especifica a primeira utilização, última utilização ou última modificação.
Valor	Significado
0 (RST_FIRSTUSED)	Primeira utilização.
1 (RST_LASTUSED)	Última utilização.
2 (RST_LASTMODIFIED)	Última modificação.

### Valor de Retorno

Zero se tiver êxito, diferente de zero se não tiver êxito. Se o valor de retorno não for zero, você pode chamar `GetLastErrorString()` ou `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` ou `GetLastSQLState()` para obter informações de erro adicionais.

### GetQMFOBJECTInfoEx()

short `GetQMFOBJECTInfoEx(BSTR ProprietarioeNome, short Tipo, short Hora)`

### Descrição

Esta função retorna informações sobre um objeto do QMF. As informações retornadas são especificadas pelos parâmetros *Tipo* e *Hora*. Após a chamada desta função, a propriedade *QMFOBJECTInfo* pode ser interrogada para obtenção do valor retornado.

### Parâmetros

Nome	Descrição
ProprietarioeNome	Uma cadeia que contém o proprietário e o nome, separados por um ponto, do objeto cujas informações você deseja recuperar. Por exemplo, Paulo.Consulta2
Tipo	Especifica o tipo de informações a obter.
Valor	Significado
0 (RSI_COMMENT)	Comentário
1 (RSI_LEVEL)	Nível
2 (RSI_TYPE)	Tipo
3 (RSI_SUBTYPE)	Subtipo
4 (RSI_RESTRICTED)	Restrito
5 (RSI_MODEL)	Modelo
6 (RSI_TIMESUSED)	Número de vezes utilizado.
7 (RSI_TIMESRUN)	Número de vezes executado.
8 (RSI_TIMESCANCELLED)	Número de vezes cancelado.
9 (RSI_DATE)	Data da primeira utilização, última utilização ou última modificação.
10 (RSI_TIME)	Hora da primeira utilização, última utilização ou última modificação.
11 (RSI_USERID)	ID do usuário da primeira utilização, última utilização ou última modificação.

12 (RSI_SQLID)	ID SQL da primeira utilização, última utilização ou última modificação.
13 (RSI_ENVIRONMENT)	Ambiente da primeira utilização, última utilização ou última modificação.
14 (RSI_MODE)	Modo da primeira utilização, última utilização ou última modificação.
15 (RSI_COMMAND)	Comando da primeira utilização, última utilização ou última modificação.
<i>Hora</i>	Especifica a primeira utilização, última utilização ou última modificação.
<b>Valor</b>	<b>Significado</b>
0 (RST_FIRSTUSED)	Primeira utilização.
1 (RST_LASTUSED)	Última utilização.
2 (RST_LASTMODIFIED)	Última modificação.

### Valor de Retorno

Zero se tiver êxito, diferente de zero se não tiver êxito. Se o valor de retorno não for zero, você pode chamar `GetLastErrorString()` ou `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` ou `GetLastSQLState()` para obter informações de erro adicionais.

### GetQMFOBJECTList()

`short GetQMFOBJECTList(BSTR Proprietário, BSTR Nome, short Tipo, VARIANT* Lista)`

#### Descrição

Esta função retorna uma matriz dos nomes de todos os objetos do QMF que correspondem aos padrões especificados nos parâmetros *Proprietário* e *Nome*.

#### Parâmetros

Nome	Descrição
Proprietário	Uma cadeia que contém o proprietário dos objetos que você deseja incluir na lista retornada.
Nome	Uma cadeia que contém o nome dos objetos que você deseja incluir na lista retornada.

Lista	Um ponteiro para uma VARIANT na qual o resultado é armazenado. O resultado é uma matriz de cadeias (tipo de variante VT_ARRAY   VT_BSTR), cada uma no formato Proprietário.Nome. Se não forem encontradas consultas correspondentes do QMF para Windows, o resultado será vazio (tipo de variante VT_EMPTY). Você deve inicializar adequadamente a VARIANT antes de chamar esta função. O Visual Basic faz isto automaticamente. Programadores do Visual C++ devem chamar VariantInit().
-------	--

**Nota:** Devido a um erro no Microsoft Excel 7.0 e no Microsoft Access 7.0 (e possivelmente em outros produtos de 32 bits da Microsoft que utilizam o Visual Basic for Applications), os dados de cadeia nas variáveis Variante recebidos do QMF para Windows podem não ser convertidos de Unicode (utilizado pelo OLE) para ANSI (utilizado pelo VBA). Quando isto ocorre, somente o primeiro caractere da cadeia é exibido. Para corrigir este problema, defina a variável como uma cadeia vazia antes de chamar a função do QMF para Windows que utiliza a variável.

Tipo	Especifica os tipos de objetos do QMF que você deseja incluir na lista. Estes valores podem ser incluídos para especificar vários tipos de objetos.
------	---

Valor	Significado
2048 (RSQ_MASK_QUERIES)	Incluir consultas do QMF na lista.
1024 (RSQ_MASK_FORMS)	Incluir formulários do QMF na lista.
512 (RSQ_MASK_PROCS)	Incluir procedimentos do QMF na lista.
256 (RSQ_MASK_TABLES)	Incluir tabelas na lista.

### Valor de Retorno

Zero se tiver êxito, diferente de zero se não tiver êxito. Se não forem encontrados objetos do QMF correspondentes, o valor de retorno será zero. Se o valor de retorno não for zero, você pode chamar GetLastErrorString() ou GetLastErrorType(), GetLastSQLCode(), GetLastSQLError() ou GetLastSQLState() para obter informações de erro adicionais.

### GetQMFOBJECTListEx()

short GetQMFOBJECTListEx(BSTR *Proprietário*, BSTR *Nome*, short *Índice*)

### Descrição

Esta função retorna o nome do objeto do QMF que corresponde aos padrões especificados nos parâmetros *Proprietário* e *Nome* referenciados pelo parâmetro

*Índice.* Após a chamada desta função, a propriedade *Value* pode ser interrogada para se obter o valor retornado.

### Parâmetros

Nome	Descrição
Proprietário	Uma cadeia que contém o proprietário dos objetos que você deseja incluir na lista retornada.
Nome	Uma cadeia que contém o nome dos objetos que você deseja incluir na lista retornada.
Índice	O índice da lista de objetos do QMF que correspondem ao padrão.
<i>Tipo</i>	Especifica os tipos de objetos do QMF que você deseja incluir na lista. Estes valores podem ser incluídos para especificar vários tipos de objetos.
Valor	Significado
2048 (RSQ_MASK_QUERIES)	Incluir consultas do QMF na lista.
1024 (RSQ_MASK_FORMS)	Incluir formulários do QMF na lista.
512 (RSQ_MASK_PROCS)	Incluir procedimentos do QMF na lista.
256 (RSQ_MASK_TABLES)	Incluir tabelas na lista.

### Valor de Retorno

Zero se tiver êxito, diferente de zero se não tiver êxito. Se não forem encontrados objetos do QMF correspondentes, o valor de retorno será `RS_ERROR_OUTOFRANGE`. Se o valor de retorno não for zero, você pode chamar `GetLastErrorString()` ou `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` ou `GetLastSQLState()` para obter informações de erro adicionais.

### GetQMFFProcText()

BSTR GetQMFFProcText(BSTR *Proprietário*e*Nome*)

### Descrição

Esta função retorna o texto que é executado para o procedimento especificado, após a substituição de variáveis. Você deve utilizar `SetProcVariable()` para definir o valor de quaisquer variáveis utilizadas no procedimento antes de chamar esta função.

### Parâmetros

Nome	Descrição
------	-----------

---

ProprietarioeNome	Uma cadeia que contém o proprietário e o nome, separados por um ponto, do objeto que você deseja excluir. Por exemplo, Paulo.Proc2
-------------------	--

---

### Valor de Retorno

Uma cadeia contendo o texto do procedimento que foi recuperado ou NULL se o procedimento não pôde ser recuperado. Se o valor de retorno for NULL, você poderá chamar `GetLastErrorString()`, `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError` ou `GetLastSQLState()` para obter informações de erro adicionais.

## GetQMFQueryText()

BSTR GetQMFQueryText(BSTR *ProprietarioeNome*)

### Descrição

Esta função recupera o texto do SQL armazenado na consulta especificada.

### Parâmetros

---

Nome	Descrição
ProprietarioeNome	Uma cadeia que contém o proprietário e o nome, separados por um ponto, do objeto que você deseja excluir. Por exemplo, Paulo.Consulta2

---

### Valor de Retorno

Uma cadeia contendo o texto da consulta que foi recuperada ou NULL se a consulta não pôde ser recuperada. Se o valor de retorno for NULL, você poderá chamar `GetLastErrorString()`, `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError` ou `GetLastSQLState()` para obter informações de erro adicionais.

## GetQueryText()

BSTR GetQueryText(long *IDConsulta*)

### Descrição

Esta função retorna o texto do SQL que é executado para a consulta especificada, após a substituição de variáveis. Você deve utilizar `SetVariable()` para definir o valor de quaisquer variáveis utilizadas na consulta antes de chamar esta função.

### Parâmetros

---

Nome	Descrição
IDConsulta	O ID da consulta, conforme retornado de <code>InitializeQuery()</code> .

---

**Nota:** O texto da consulta não está disponível para instruções SQL estáticas. Para IDs de consultas retornados de `InitializeStaticQuery()`, `GetQueryText()` retorna uma cadeia vazia.

### Valor de Retorno

Se tiver êxito, uma cadeia contendo o texto do SQL é retornada. Se não tiver êxito, NULL é retornado. Se o valor de retorno for NULL, você pode chamar `GetLastErrorString()` ou `GetLastErrorType()` para obter informações de erro adicionais.

## GetQueryVerb()

BSTR GetQueryVerb(long *IDConsulta*)

### Descrição

Esta função retorna uma cadeia que contém o verbo SQL utilizado na consulta.

### Parâmetros

Nome	Descrição
IDConsulta	O ID da consulta, conforme retornado de <code>InitializeQuery()</code> .

**Nota:** O verbo da consulta não está disponível para instruções SQL estáticas. Para IDs de consultas retornados de `InitializeStaticQuery()`, `GetQueryVerb()` retorna uma cadeia vazia.

### Valor de Retorno

Se tiver êxito, uma cadeia contendo o verbo SQL é retornada. Se não tiver êxito, NULL é retornado. Se o valor de retorno for NULL, você pode chamar `GetLastErrorString()` ou `GetLastErrorType()` para obter informações de erro adicionais.

## GetResourceLimit()

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

### Descrição

Esta função obtém o limite do recurso solicitado. Você deve chamar `InitializeServer()` antes de chamar esta função, pois os limites de recursos são tratados individualmente para cada servidor.

### Parâmetros

Nome	Descrição
Recurso	Os valores de recursos incluem:

Valor	Significado
0 (RSR_IDLE_CONNECTION_TIMEOUT)	Tempo limite de conexão inativa (em segundos).
1 (RSR_IDLE_QUERY_TIMEOUT_CANCEL)	Tempo limite de consulta inativa (em segundos).
2 (RSR_IDLE_QUERY_TIMEOUT_WARNING)	Tempo limite de consulta inativa (em segundos). Este é o limite do aviso; ele não é imposto para a API do QMF para Windows.
3 (RSR_SERVER_RESPONSE_TIMEOUT_CANCEL)	Tempo limite do servidor (em segundos).
4 (RSR_SERVER_RESPONSE_TIMEOUT_WARNING)	Tempo limite do servidor (em segundos). Este é o limite do aviso; ele não é imposto para a API do QMF para Windows.
5 (RSR_MAX_ROWS_TO_FETCH_CANCEL)	Número máximo de linhas a serem buscadas.
6 (RSR_MAX_ROWS_TO_FETCH_WARNING)	Número máximo de linhas a serem buscadas. Este é o limite do aviso; ele não é imposto para a API do QMF para Windows.
7 (RSR_MAX_BYTES_TO_FETCH_CANCEL)	Número máximo de bytes a serem buscados.
8 (RSR_MAX_BYTES_TO_FETCH_WARNING)	Número máximo de bytes a serem buscados. Este é o limite do aviso; ele não é imposto para a API do QMF para Windows.
9 (RSR_MAX_CONNECTIONS)	Número máximo de conexões permitidas com o servidor do banco de dados.
10 (RSR_ALLOW_SERVER_ACCESS_UI)	É permitido acesso ao servidor do banco de dados a partir da interface do QMF para Windows?
11 (RSR_ALLOW_SERVER_ACCESS_API)	É permitido acesso ao servidor do banco de dados a partir da API do QMF para Windows?

12 (RSR_FETCH_ALL_ROWS)	Buscar automaticamente todas as linhas?
13 (RSR_CONFIRM_UPDATES)	Confirmar atualizações do servidor do banco de dados? Esta opção não tem efeito sobre a API do QMF para Windows; as atualizações do banco de dados não são confirmadas para a API do QMF para Windows.
14 (RSR_SUMMARY_TRACKING)	O rastreamento de objeto resumido está ativado?
15 (RSR_DETAILED_TRACKING)	O rastreamento de objeto detalhado está ativado?
16 (RSR_SQL_TRACKING)	O rastreamento de texto de SQL está ativado?
17 (RSR_ADHOC_TRACKING)	O rastreamento de objeto ad hoc está ativado?
18 (RSR_ALLOW_ACQUIRE)	O verbo ACQUIRE do SQL é permitido?
19 (RSR_ALLOW_ALTER)	O verbo ALTER do SQL é permitido?
20 (RSR_ALLOW_COMMENT)	O verbo COMMENT do SQL é permitido?
21 (RSR_ALLOW_CREATE)	O verbo CREATE do SQL é permitido?
22 (RSR_ALLOW_DELETE)	O verbo DELETE do SQL é permitido?
23 (RSR_ALLOW_DROP)	O verbo DROP do SQL é permitido?
24 (RSR_ALLOW_EXPLAIN)	O verbo EXPLAIN do SQL é permitido?
25 (RSR_ALLOW_GRANT)	O verbo GRANT do SQL é permitido?
26 (RSR_ALLOW_INSERT)	O verbo INSERT do SQL é permitido?
27 (RSR_ALLOW_LABEL)	O verbo LABEL do SQL é permitido?
28 (RSR_ALLOW_LOCK)	O verbo LOCK do SQL é permitido?

29 (RSR_ALLOW_REVOKE)	O verbo REVOKE do SQL é permitido?
30 (RSR_ALLOW_SELECT)	O verbo SELECT do SQL é permitido?
31 (RSR_ALLOW_SET)	O verbo SET do SQL é permitido?
32 (RSR_ALLOW_SIGNAL)	O verbo SIGNAL do SQL é permitido?
33 (RSR_ALLOW_UPDATE)	O verbo UPDATE do SQL é permitido?
34 (RSR_ALLOW_CALL)	O verbo CALL do SQL é permitido?
35 (RSR_ALLOW_SAVE_DATA)	O comando Salvar Dados é permitido?
36 (RSR_SAVE_DATA_TABLE_SPACE_NAME)	O nome padrão da coleção para ligar pacotes?
37 (RSR_SAVE_DATA_TABLE_SPACE_NAME_OVERRIDE)	O nome de espaço de tabela padrão para o comando Salvar Dados pode ser substituído pelo usuário?
38 (RSR_ALLOW_BIND_PACKAGE)	Permitir ligação de pacotes?
39 (RSR_DEF_COLLECTION)	O nome da coleção padrão para ligação de pacotes.
40 (RSR_DEF_COLLECTION_OVERRIDE)	O nome da coleção padrão para ligação de pacotes pode ser substituído pelo usuário?
41 (RSR_DEF_ISOLATION_LEVEL)	O nível de isolamento padrão para ligação de pacotes.
42 (RSR_DEF_ISOLATION_LEVEL_OVERRIDE)	O nível de isolamento padrão para ligação de pacotes pode ser substituído pelo usuário?
43 (RSR_ALLOW_TABLE_EDIT)	Permitir utilização do editor de tabela?
44 (RSR_ALLOW_EXPORT)	Permitir exportação de dados?

45 (RSR_ALLOW_SAVED_QUERIES_ONLY)	O usuário tem permissão para executar somente consultas salvas?
46 (RSR_ALLOW_DROP_PACKAGE)	Permitir eliminação de pacotes?
47 (RSR_QUERY_ISOLATION_LEVEL)	O nível de isolamento a ser utilizado ao executar consultas.
48 (RSR_ACCOUNT_STRING)	A cadeia que especifica informações de contas a serem passadas na conexão com o servidor do banco de dados.
49 (RSR_ACCOUNT_OVERRIDE)	A cadeia que especifica informações a serem passadas na conexão com o servidor do banco de dados pode ser substituída pelo usuário?
Valor	Um ponteiro para um inteiro longo no qual o resultado é armazenado. O resultado é o valor do limite de recursos solicitado. Para valores booleanos, o resultado é diferente de zero para verdadeiro, zero para falso. Para RSR_SAVE_DATA_TABLE_SPACE_NAME, RSR_DEF_COLLECTION e RSR_ACCOUNT_STRING, -1 é retornado e a propriedade <i>ResourceLimit</i> pode ser interrogada para obtenção do valor da cadeia retornado.

### Valor de Retorno

Zero se tiver êxito, diferente de zero se não tiver êxito. Se o valor de retorno for diferente de zero, você pode chamar `GetLastErrorString()` ou `GetLastErrorType()` para obter informações de erro adicionais.

### GetResourceLimitEx()

`short GetResourceLimitEx(short Recurso)`

### Descrição

Esta função obtém o limite do recurso solicitado. Você deve chamar `InitializeServer()` antes de chamar esta função, pois os limites de recursos são tratados individualmente para cada servidor. Após uma chamada para esta função, faça uma consulta na propriedade *ResourceLimit* para obter o resultado.

## Parâmetros

Nome	Descrição
Recurso	Os valores de recursos são os mesmos que os da chamada <code>GetResourceLimit()</code> .

**Nota:** O verbo da consulta não está disponível para instruções SQL estáticas. Para IDs de consultas retornados de `InitializeStaticQuery()`, `GetQueryVerb()` retorna uma cadeia vazia.

### Valor de Retorno

Zero se tiver êxito, diferente de zero se não tiver êxito. Se o valor de retorno for diferente de zero, você pode chamar `GetLastErrorString()` ou `GetLastErrorType()` para obter informações de erro adicionais.

## GetRowCount()

`long GetRowCount(long IDConsulta)`

### Descrição

Esta função retorna o número de linhas atualmente no buffer interno do QMF para Windows. Ele pode ser maior que o número de linhas recuperadas com `FetchNextRow()` ou `FetchNextRows()`, pois o QMF para Windows armazena em buffer os dados recebidos do banco de dados.

Esta função retorna o número de linhas já recuperadas do banco de dados. Se você quiser recuperar o número total de linhas no conjunto de resultados, pode:

- Chamar `CompleteQuery()` e buscar todas as linhas utilizando `FetchNextRow()` ou `FetchNextRows()`.
- Especificar `BuscarTodasLinhas = TRUE` quando chamar `Open()`.

## Parâmetros

Nome	Descrição
<i>IDConsulta</i>	O ID da consulta, conforme retornado de <code>InitializeQuery()</code> .

### Valor de Retorno

O número de linhas se tiver êxito (0 se não foram recuperadas linhas) ou -1 se não tiver êxito. Se for 1, você pode chamar `GetLastErrorString()` ou `GetLastErrorType()` para obter informações de erro adicionais.

## GetServerList()

`short GetServerList(VARIANT* Lista)`

## Descrição

Esta função retorna uma matriz que contém os nomes dos servidores de banco de dados definidos no Arquivo de Definição do Servidor (SDF, Server Definition File) do QMF para Windows. Você deve definir um servidor de banco de dados no arquivo SDF se desejar acessá-lo utilizando a API do QMF para Windows.

## Parâmetros

Nome	Descrição
Lista	Um ponteiro para uma VARIANT na qual o resultado é armazenado. O resultado é uma matriz de cadeias (tipo de variante VT_ARRAY   VT_BSTR) com cada cadeia contendo o nome de um servidor de banco de dados. Se você não tiver definido um servidor de banco de dados, o resultado será vazio (tipo de variante VT_EMPTY). Você deve inicializar adequadamente a VARIANT antes de chamar esta função. O Visual Basic faz isto automaticamente. Programadores do Visual C++ devem chamar VariantInit().

**Nota:** Devido a um erro no Microsoft Excel 7.0 e no Microsoft Access 7.0 (e possivelmente em outros produtos de 32 bits da Microsoft que utilizam o Visual Basic for Applications), os dados de cadeia nas variáveis Variante recebidos do QMF para Windows podem não ser convertidos de Unicode (utilizado pelo OLE) para ANSI (utilizado pelo VBA). Quando isto ocorre, somente o primeiro caractere da cadeia é exibido. Para corrigir este problema, defina a variável como uma cadeia vazia antes de chamar a função do QMF para Windows que utiliza a variável.

## Valor de Retorno

Zero se tiver êxito, diferente de zero se não tiver êxito. Se você não tiver definido um servidor de banco de dados, o valor de retorno será zero. Se o valor de retorno for diferente de zero, você pode chamar GetLastErrorString() ou GetLastErrorType() para obter informações de erro adicionais.

## GetServerListEx()

short GetServerListEx(short *Índice*)

## Descrição

Esta função recupera o nome do servidor referenciado pelo parâmetro *Índice*. Após a chamada desta função, a propriedade *Value* pode ser interrogada para se obter o valor retornado.

## Parâmetros

Nome	Descrição
------	-----------

---

Índice	Um índice para a lista de servidores.
--------	---------------------------------------

---

### Valor de Retorno

Zero se tiver êxito, `RS_OUTOFRANGE` quando o índice for maior que o número de servidores disponíveis, diferente de zero se não tiver êxito. Se você não tiver definido servidores de banco de dados, o valor de retorno será `RS_OUTOFRANGE`. Se o valor de retorno for diferente de zero, você pode chamar `GetLastErrorString()` ou `GetLastErrorType()` para obter informações de erro adicionais.

## GetStoredProcedureResultSets()

short GetStoredProcedureResultSets(long *IDConsulta*, VARIANT\* *ConjuntosResultados*)

### Descrição

Esta função recupera os IDs de consulta para os conjuntos de resultados retornados pelo procedimento armazenado executado com o *IDConsulta* original. Cada ID de consulta retornado pode ser utilizado com `FetchNextRow()` ou `FetchNextRows()` para recuperar as linhas do conjunto de resultados e `Close()` quando o final de cada conjunto de resultados for alcançado.

### Parâmetros

---

Nome	Descrição
<i>IDConsulta</i>	O ID da consulta original, conforme retornado de <code>InitializeQuery()</code> .
<i>ConjuntosResultados</i>	Um ponteiro para uma VARIANT na qual os IDs de consultas para os conjuntos de resultados são armazenados. O resultado é uma matriz de inteiros longos (tipo de variante <code>VT_ARRAY   VT_I4</code> ), com cada inteiro sendo o ID da consulta para os conjuntos de resultados correspondentes. Se o procedimento armazenado não retornou nenhum conjunto de resultados, o resultado será vazio (tipo de variante <code>VT_EMPTY</code> ). Você deve inicializar adequadamente a VARIANT antes de chamar esta função. O Visual Basic faz isto automaticamente. Programadores do Visual C++ devem chamar <code>VariantInit()</code> .

---

### Valor de Retorno

Zero se tiver êxito, diferente de zero se não tiver êxito. Se o valor de retorno não for zero, você pode chamar `GetLastErrorString()` ou `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` ou `GetLastSQLState()` para obter informações de erro adicionais.

## GetVariables()

short GetVariables(long *IDConsulta*, VARIANT\* *Variáveis*)

### Descrição

Esta função retorna uma matriz dos nomes das variáveis no texto SQL da consulta. Você deve atribuir valores a essas variáveis chamando SetVariable() antes de executar a consulta utilizando Open() ou Execute().

### Parâmetros

Nome	Descrição
IDConsulta	O ID da consulta, conforme retornado de InitializeQuery().
Variáveis	Um ponteiro para uma VARIANT na qual o resultado é armazenado. O resultado é uma matriz de cadeias (tipo de variante VT_ARRAY   VT_BSTR) com cada cadeia contendo o nome de uma variável. Se não existirem variáveis na instrução SQL, o resultado será vazio (tipo de variante VT_EMPTY). Você deve inicializar adequadamente a VARIANT antes de chamar esta função. O Visual Basic faz isto automaticamente. Programadores do Visual C++ devem chamar VariantInit().

**Nota:** Devido a um erro no Microsoft Excel 7.0 e no Microsoft Access 7.0 (e possivelmente em outros produtos de 32 bits da Microsoft que utilizam o Visual Basic for Applications), os dados de cadeia nas variáveis Variante recebidos do QMF para Windows podem não ser convertidos de Unicode (utilizado pelo OLE) para ANSI (utilizado pelo VBA). Quando isto ocorre, somente o primeiro caractere da cadeia é exibido. Para corrigir este problema, defina a variável como uma cadeia vazia antes de chamar a função do QMF para Windows que utiliza a variável.

### Valor de Retorno

Zero se tiver êxito, diferente de zero se não tiver êxito. Se não existirem variáveis na instrução SQL, o valor de retorno será RS\_NO\_ERROR\_NO\_DATA (-1). Se o valor de retorno for diferente de zero, você pode chamar GetLastErrorString() ou GetLastErrorType() para obter informações de erro adicionais.

## GetVariablesEx()

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

### Descrição

Esta função retorna o nome da variável no texto SQL da consulta referenciada pelo parâmetro *Índice*. Após a chamada desta função, a propriedade *Value* pode ser interrogada para se obter o valor retornado. Você deve atribuir

valores a esta variável (e a todas as outras no texto SQL) chamando SetVariable() antes de executar a consulta utilizando Open() ou Execute().

### Parâmetros

Nome	Descrição
IDConsulta	O ID da consulta, conforme retornado de InitializeQuery().
Variáveis	Um índice para a lista interna de variáveis. Faça uma consulta à propriedade <i>Value</i> para a cadeia que corresponde ao índice passado. Se não existirem variáveis na instrução SQL, a função retornará RS_NO_ERROR_NO_DATA (-1).

### Valor de Retorno

Zero se tiver êxito, diferente de zero se não tiver êxito. Se não existirem variáveis na instrução SQL, o valor de retorno será RS\_NO\_ERROR\_NO\_DATA (-1). Se o valor de retorno for diferente de zero, você pode chamar GetLastErrorString() ou GetLastErrorType() para obter informações de erro adicionais.

### InitializeProc()

long InitializeProc(short *TipoOrigem*, BSTR *Origem*)

### Descrição

Esta função define o texto que você deseja utilizar em um procedimento. Você pode passar o texto como um parâmetro para esta função, lê-lo de um arquivo de texto ou obtê-lo de um procedimento existente.

### Parâmetros

Nome	Descrição
TipoOrigem	Especifica a origem para o texto do procedimento.

Valor	Significado
0 (RSS_STRING)	O texto está contido no parâmetro <i>Origem</i> .
2 (RSS_FILE)	O texto está contido no arquivo de texto cujo nome está especificado pelo parâmetro <i>Origem</i> .
3 (RSS_QMFPROC)	O texto está contido no procedimento cujo proprietário e nome estão especificados pelo parâmetro <i>Origem</i> .

<i>Origem</i>	Uma cadeia que contém o texto, o proprietário e nome (Proprietário.Nome) do procedimento ou o nome de um arquivo que contém o texto do procedimento.
---------------	--

### Valor de Retorno

Se tiver êxito, o ID do procedimento (IDProc). Se não tiver êxito, -1. Você deve utilizar este valor em todas as chamadas da interface que requerem o parâmetro *IDProc*.

### InitializeQuery()

long InitializeQuery(short *TipoOrigem*, BSTR *Origem*)

#### Descrição

Esta função define o texto que você deseja utilizar em uma consulta. Você pode passar o texto do SQL como um parâmetro para esta função, lê-lo de um arquivo de texto ou obtê-lo de uma consulta existente. Chame close() quando tiver terminado a consulta.

#### Parâmetros

Nome	Descrição
TipoOrigem	Especifica a origem para o texto da instrução SQL.

Valor	Significado
0 (RSS_STRING)	O texto do SQL está contido no parâmetro <i>Origem</i> .
1 (RSS_QMFQUERY)	O texto do SQL está contido na consulta cujo proprietário e nome estão especificados pelo parâmetro <i>Origem</i> .
2 (RSS_FILE)	O texto do SQL está contido no arquivo de texto cujo nome está especificado pelo parâmetro <i>Origem</i> .

### Valor de Retorno

Se tiver êxito, o ID da consulta. Se não tiver êxito, -1. Você deve utilizar este valor em todas as chamadas da interface que requerem o parâmetro *Consulta*.

### InitializeServer()

short InitializeServer(BSTR *NomeServidor*, BSTR *IDUsuário*, BSTR *Senha*, BOOL *ForçarDiálogo*, [VARIANT *Conta*], [VARIANT *SuprimirDiálogo*])

#### Descrição

Esta função inicializa uma conexão com um servidor de banco de dados. Você deve chamá-la antes de chamar qualquer outra função na API do QMF para Windows. Você pode chamar esta função várias vezes. Entretanto, se chamá-la e não terminar chamando Commit() ou Rollback(), ocorrerá uma reversão implícita.

## Parâmetros

Nome	Descrição
NomeServidor	Uma cadeia que contém o nome do servidor de banco de dados que você deseja utilizar. Este nome deve corresponder a um dos nomes definidos no Arquivo de Definição de Servidor do QMF para Windows. Chame <code>GetServerList()</code> para recuperar uma lista de servidores válidos.
IDUsuário	Uma cadeia que contém o ID do usuário que você deseja utilizar. Se <code>IDUsuário</code> for <code>NULL</code> ou uma cadeia vazia, o QMF para Windows tentará utilizar o ID do usuário da consulta mais recente, se estiver disponível. Caso contrário, o QMF para Windows exibirá a caixa de diálogo Informações do Usuário para obter um ID de usuário e senha.
Senha	Uma cadeia que contém a senha para o ID do usuário especificado. Se uma Senha for <code>NULL</code> ou uma cadeia vazia, o QMF para Windows tentará utilizar uma senha memorizada, se estiver disponível (requer o Windows para Workgroups). Se não houver senha disponível, o QMF para Windows exibirá a caixa de diálogo Informações do Usuário para obter uma senha.
ForçarDiálogo	Um valor diferente de zero indica que o QMF para Windows exibe a caixa de diálogo Informações do Usuário independentemente de um ID do Usuário e Senha terem sido especificados. Isto dá ao usuário uma oportunidade de alterar as informações antes de serem utilizadas. Zero indica que o QMF para Windows deve exibir a caixa de diálogo Informações do Usuário somente quando necessário.
Conta	Opcionalmente, uma cadeia que especifica informações de conta a serem passadas ao servidor na conexão. O servidor pode utilizar essas informações em um sistema de contabilização de jobs.
SuprimirDiálogo	Um valor diferente de zero indica que o QMF para Windows não exibe a caixa de diálogo Informações do Usuário, mesmo se não tiverem sido especificados um ID do usuário e uma senha. Esta opção é útil na execução em um ambiente onde não haja usuário presente para responder à caixa de diálogo Informações do Usuário; por exemplo, em um servidor da Web.

### Valor de Retorno

Zero se tiver êxito, diferente de zero se não tiver êxito. Se o valor de retorno não for zero, você pode chamar `GetLastErrorString()` ou `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` ou `GetLastSQLState()` para obter informações de erro adicionais.

## Tópicos Relacionados

SetParent()

### InitializeStaticQuery()

long InitializeStaticQuery(BSTR NomeColeção, BSTR NomePacote, BSTR TokenConsistência, short NúmeroSeção)

#### Descrição

Esta função especifica a seção de um pacote que você deseja executar como uma consulta estática.

#### Parâmetros

Nome	Descrição
NomeColeção	O nome de uma coleção anteriormente ligada.
NomePacote	O nome de um pacote anteriormente ligado.
TokenConsistência	O token utilizado pela coleção e pacote denominados acima.
NúmeroSeção	O número da seção da instrução na coleção e pacote que você deseja executar.

#### Valor de Retorno

Se tiver êxito, o ID da consulta. Se não tiver êxito, -1. Você deve utilizar este valor em todas as chamadas de interface que requeram o parâmetro *IDConsulta*.

### IsStatic()

BOOL IsStatic(long IDConsulta)

#### Descrição

Esta função determina se o ID da consulta especificado refere-se a uma consulta estática ou dinâmica.

#### Parâmetros

Nome	Descrição
IDConsulta	O ID da consulta, conforme retornado de InitializeQuery() ou InitializeStaticQuery().

#### Valor de Retorno

Retorna um valor diferente de zero se tiver êxito e *IDConsulta* refere-se a uma consulta estática, ou zero caso contrário.

### Open()

short Open(long IDConsulta, long LimiteLinhas, BOOL BuscarTodasLinhas)

## Descrição

Utilize esta função para executar uma consulta que utiliza o verbo SELECT, abrindo um cursor no banco de dados para a consulta. Utilize `FetchNextRow()` ou `FetchNextRows()` para recuperar os dados para a consulta e chame `Close()` quando terminar. Se o QMF para Windows estiver configurado para buscar automaticamente todas as linhas (consulte `RSR_FETCHALLROWS` na descrição para `GetResourceLimit()`) ou o parâmetro `BuscarTodasLinhas` não for zero, o QMF para Windows buscará todas as linhas do conjunto de resultados em seu buffer interno antes de retornar desta chamada.

**Nota:** O nome desta função entra em conflito com a palavra-chave `Open` do Microsoft Access 2.0. Se você estiver utilizando o MS Access 2.0, coloque o nome da função entre colchetes [ ].

**Nota:** Utilize esta função somente em instruções que contenham o verbo SELECT do SQL. Para instruções que contenham qualquer outro verbo (por exemplo, SET), chame `Execute()`. Para determinar o verbo utilizado por uma consulta, chame `GetQueryVerb()`.

## Parâmetros

Nome	Descrição
<i>IDConsulta</i>	O ID da consulta, conforme retornado de <code>InitializeQuery()</code> .
<i>LimiteLinhas</i>	Um número indicando o número máximo de linhas a serem recuperadas do banco de dados. Zero indica que não há limite obrigatório, exceto pelo limite de linhas estabelecido pelo programa Administrador do QMF para Windows.
<i>BuscarTodasLinhas</i>	Um valor booleano que indica se todas as linhas do conjunto de resultados são buscadas automaticamente ou não para o buffer interno do QMF para Windows. Se for diferente de zero, todas as linhas serão buscadas automaticamente, fechando o cursor e liberando o banco de dados para utilização por outros usuários. Isto é o mesmo que chamar <code>CompleteQuery()</code> .

## Valor de Retorno

Zero se tiver êxito, diferente de zero se não tiver êxito. Se o valor de retorno não for zero, você pode chamar `GetLastErrorString()` ou `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` ou `GetLastSQLState()` para obter informações de erro adicionais.

## Prepare()

`short Prepare(long IDConsulta)`

### Descrição

Esta função prepara a consulta especificada por *IDConsulta*. A instrução é examinada pelo servidor do banco de dados, verificando a existência do objeto, autorizações requeridas, etc. Se a consulta for uma instrução *SELECT*, as informações sobre as colunas retornadas pela instrução estarão disponíveis após a conclusão de *Prepare()*.

### Parâmetros

Nome	Descrição
<i>IDConsulta</i>	O ID da consulta, conforme retornado de <i>InitializeQuery()</i> .

### Valor de Retorno

Zero se tiver êxito, diferente de zero se não tiver êxito. Se o valor de retorno não for zero, você pode chamar *GetLastErrorString()* ou *GetLastErrorType()*, *GetLastSQLCode()*, *GetLastSQLError()* ou *GetLastSQLState()* para obter informações de erro adicionais.

### Tópicos Relacionados

*Execute()*

*Open()*

### PrintReport()

*short PrintReport(long IDConsulta, short TipoOrigem, BSTR Origem, BSTR NomeArquivoSaída, short ComprimentoPágina, short LarguraPágina, BOOL IncluirDataHora, BOOL IncluirNúmPágina, [VARIANT Formato], [VARIANT UtilizarConfPáginaFormulário])*

### Descrição

*PrintReport()* é um sinônimo para *ExportReport()*.

### ReinitializeServer()

*short ReinitializeServer()*

### Descrição

Esta função reinicializa a conexão com um servidor do banco de dados. Normalmente, você precisará chamar esta função apenas se uma das outras funções da API do QMF para Windows retornar um erro. A chamada desta função resulta em uma reversão implícita, que fecha todos os cursores abertos e invalida todos os IDs de consultas pendentes.

### Valor de Retorno

Zero se tiver êxito, diferente de zero se não tiver êxito. Se o valor de retorno não for zero, você pode chamar *GetLastErrorString()* ou *GetLastErrorType()*, *GetLastSQLCode()*, *GetLastSQLError()* ou *GetLastSQLState()* para obter informações de erro adicionais.

## Rollback()

short Rollback()

### Descrição

Esta função cancela quaisquer alterações feitas na unidade de trabalho atual, termina a unidade de trabalho atual, fecha todos os cursores abertos e invalida todos os IDs de Consultas pendentes.

**Nota:** O nome desta função entra em conflito com a palavra-chave Rollback do Microsoft Access 2.0. Se você estiver utilizando o MS Access 2.0, coloque o nome da função entre colchetes [ ].

**Nota:** A reversão afeta apenas alterações de SQL que foram executadas chamando-se Open() ou Execute(). Ela não afeta alterações feitas por outras funções da API do QMF para Windows, como FastSaveData(), SaveData() ou DeleteQMFObject().

### Valor de Retorno

Zero se tiver êxito, diferente de zero se não tiver êxito. Se o valor de retorno não for zero, você pode chamar GetLastErrorString() ou GetLastErrorType(), GetLastErrorSQLCode(), GetLastErrorSQLError() ou GetLastErrorSQLState() para obter informações de erro adicionais.

### Tópicos Relacionados

Commit()

## RunProc()

short RunProc(long *IDProc*)

### Descrição

Esta função executa o procedimento especificado. O procedimento é executado até a conclusão ou até ocorrer um erro. Não é possível acessar nenhum dos resultados do procedimento (por exemplo, dados de uma consulta que foi executada) através desta interface de programação. Entretanto, quaisquer arquivos exportados ou dados salvos pelo procedimento estarão disponíveis após a execução.

### Parâmetros

Nome	Descrição
<i>IDProc</i>	O ID do procedimento, conforme retornado de InitializeProc().

### Valor de Retorno

Zero se tiver êxito, diferente de zero se não tiver êxito. Se o valor de retorno não for zero, você pode chamar `GetLastErrorString()` ou `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` ou `GetLastSQLState()` para obter informações de erro adicionais.

### SaveData()

`short SaveData(long IDConsulta, long PrimeiraLinha, long PrimeiraCol, long ÚltimaLinha, long ÚltimaCol, BOOL Substituir, BSTR NomeTabela, BSTR NomeEspaçoTabela, BSTR NomeServidor, BSTR IDUsuário, BSTR Senha, BOOL ForçarDiálogo, [VARIANT Conta], [VARIANT Comentário], [VARIANT ConsolidarEscopo])`

### Descrição

Esta função salva o intervalo de linhas e colunas especificado na tabela especificada no espaço de tabela especificado. Você deve chamar `CompleteQuery()` antes de chamar esta função se não tiver recuperado dados de linhas para todas as linhas que deseja salvar na tabela. Se você tentar salvar linhas que não foram recuperadas do banco de dados, o salvamento falhará. Se a tabela já existir, os novos dados deverão ter o mesmo número e tipos de colunas que a tabela existente.

Esta função opera em uma unidade de trabalho separada de outras funções da API e seus resultados são automaticamente consolidados. A chamada de `Commit()` ou `Rollback()` não tem efeito sobre as alterações feitas utilizando esta função.

### Parâmetros

Nome	Descrição
<i>IDConsulta</i>	O ID da consulta, conforme retornado de <code>InitializeQuery()</code> .
<i>PrimeiraLinha</i>	A primeira linha que você deseja incluir no salvamento. O valor da primeira linha em um conjunto de resultados é 0.
<i>PrimeiraCol</i>	A primeira coluna que você deseja incluir no salvamento. O valor da primeira coluna em um conjunto de resultados é zero.
<i>ÚltimaLinha</i>	A última linha que você deseja incluir no salvamento ou -1 se todas as linhas forem incluídas. O valor da última linha em um conjunto de resultados é um a menos que o número total de linhas.
<i>ÚltimaCol</i>	A última coluna que você deseja incluir no salvamento ou -1 se todas as colunas forem incluídas. O valor da última coluna em um conjunto de resultados é um a menos que o número total de colunas.

Substituir	Um valor diferente de zero indica que os dados especificados substituem quaisquer dados existentes na tabela. Zero indica que os dados especificados são anexados a quaisquer dados existentes na tabela.
NomeTabela	O nome da tabela na qual os dados são armazenados. Se a tabela não existir, ela será criada.
NomeEspaçoTabela	O nome do espaço da tabela no qual a tabela existe ou é criada. Se <i>NomeEspaçoTabela</i> for NULL ou uma cadeia vazia, será utilizado o espaço de tabela padrão. Se você tiver configurado o QMF para Windows para sempre utilizar o espaço de tabela padrão (consulte RSR_SDDIFFERENTTS na descrição de GetResourceLimit()), este parâmetro será ignorado.
NomeServidor	O nome do servidor do banco de dados no qual a tabela está armazenada. Se <i>NomeServidor</i> for NULL ou uma cadeia vazia, o servidor especificado na chamada de InitializeServer() será utilizado e <i>IDUsuário</i> , <i>Senha</i> , <i>ForçarDiálogo</i> e <i>Conta</i> serão ignorados.
<i>IDUsuário</i>	Se você especificou um servidor diferente em <i>NomeServidor</i> , <i>IDUsuário</i> será o ID do usuário para esse servidor. Se você não especificar um ID do usuário, o QMF para Windows utilizará o último ID de usuário especificado para este servidor, se estiver disponível, ou exibirá uma caixa de diálogo se não houver um disponível. Este parâmetro será ignorado se <i>NomeServidor</i> for NULL ou uma cadeia vazia.
Senha	Se você especificou um servidor diferente em <i>NomeServidor</i> , <i>Senha</i> será a senha utilizada para esse servidor. Se você não especificar uma senha, o QMF para Windows utilizará a última senha especificada para este servidor, se estiver disponível, ou exibirá uma caixa de diálogo se não houver uma disponível. Este parâmetro será ignorado se <i>NomeServidor</i> for NULL ou uma cadeia vazia.
ForçarDiálogo	Se você especificou um servidor diferente em <i>NomeServidor</i> , um valor diferente de zero força o QMF para Windows a exibir uma caixa de diálogo solicitando informações de início de sessão, mesmo se forem especificados um ID de usuário e senha ou estes estiverem disponíveis de alguma forma. Zero indica que o QMF para Windows exibe esta caixa de diálogo apenas se necessário. Este parâmetro será ignorado se <i>NomeServidor</i> for NULL ou uma cadeia vazia.
Conta	Se você especificou um servidor diferente em <i>NomeServidor</i> , opcionalmente, uma cadeia especificando informações de conta a serem passadas a esse servidor durante a conexão. O servidor pode utilizar essas informações em um sistema de contabilização de jobs. Este parâmetro será ignorado se <i>NomeServidor</i> for NULL ou uma cadeia vazia.

<i>Comentário</i>	Opcionalmente, uma cadeia que especifica um comentário para a tabela na qual os dados são salvos.
<i>ConsolidarEscopo</i>	Opcionalmente, quantas linhas serão inseridas na tabela por vez antes de consolidar a unidade de trabalho. Especificar zero indica que todas as linhas devem ser inseridas antes da consolidação. Especificar 10 (por exemplo), indica que uma consolidação deve ser realizada após cada dez linhas inseridas.

### Valor de Retorno

Zero se tiver êxito, diferente de zero se não tiver êxito. Se o valor de retorno não for zero, você pode chamar `GetLastErrorString()` ou `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` ou `GetLastSQLState()` para obter informações de erro adicionais. Se o conjunto de resultados for vazio ou não forem recuperadas linhas do banco de dados, um valor diferente de zero será retornado, exceto se `PrimeiraLinha=0` e `ÚltimaLinha=-1`. Neste caso, zero será retornado e uma tabela vazia será criada.

## SaveQMFPProc()

short SaveQMFPProc(BSTR *ProprietárioeNome*, BSTR *Texto*, BSTR *Comentário*, BOOL *Substituir*, BOOL *Compartilhar*)

### Descrição

Esta função salva um procedimento em um servidor de banco de dados.

### Parâmetros

Nome	Descrição
<i>ProprietárioeNome</i>	Uma cadeia que contém o proprietário e o nome, separados por um ponto, do procedimento que você deseja salvar. Por exemplo, Paulo.Proc2
<i>Texto</i>	Uma cadeia que contém o texto que você deseja salvar no procedimento.
<i>Comentário</i>	Uma cadeia que contém qualquer comentário que você queira salvar com o procedimento. Se não houver comentário, passe este parâmetro como uma cadeia vazia ou NULL.
<i>Substituir</i>	Um valor diferente de zero substitui um procedimento existente com o mesmo nome. Zero aborta a operação se houver um procedimento com o mesmo nome.
<i>Compartilhar</i>	Um valor diferente de zero compartilha o procedimento com outros usuários. Zero não compartilha o procedimento com outros usuários.

### Valor de Retorno

Zero se tiver êxito, diferente de zero se não tiver êxito. Se o valor de retorno não for zero, você pode chamar `GetLastErrorString()` ou `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` ou `GetLastSQLState()` para obter informações de erro adicionais.

## SaveQMFQuery()

short SaveQMFQuery(BSTR *ProprietarioeNome*, BSTR *Texto*, BSTR *Comentário*, BOOL *Substituir*, BOOL *Compartilhar*)

### Descrição

Esta função salva uma consulta em um servidor de banco de dados.

### Parâmetros

Nome	Descrição
<i>ProprietarioeNome</i>	Uma cadeia que contém o proprietário e o nome, separados por um ponto, da consulta que você deseja salvar. Por exemplo, Paulo.Consulta2
<i>Texto</i>	Uma cadeia que contém o texto que você deseja salvar na consulta.
<i>Comentário</i>	Uma cadeia que contém qualquer comentário que você queira salvar com a consulta. Se não houver comentário, passe este parâmetro como uma cadeia vazia ou NULL.
<i>Substituir</i>	Um valor diferente de zero substitui uma consulta existente com o mesmo nome. Zero aborta a operação se houver uma consulta com o mesmo nome.
<i>Compartilhar</i>	Um valor diferente de zero compartilha a consulta com outros usuários. Zero não compartilha a consulta com outros usuários.

### Valor de Retorno

Zero se tiver êxito, diferente de zero se não tiver êxito. Se o valor de retorno não for zero, você pode chamar `GetLastErrorString()` ou `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` ou `GetLastSQLState()` para obter informações de erro adicionais.

## SetBindOption()

short SetBindOption(BSTR *NomeColeção*, BSTR *NomePacote*, short *Opção*, short *Valor*)

### Descrição

Esta função define opções para a coleção e o pacote antes da chamada de `EndBind()`.

## Parâmetros

Nome	Descrição
NomeColeção	O ID da coleção do pacote para o qual você deseja definir a opção.
NomePacote	O nome do pacote para o qual você deseja definir a opção.
Opção	Uma das opções listadas abaixo.
Valor	Um valor diferente de zero substitui uma consulta existente com o mesmo nome. Zero aborta a operação se houver uma consulta com o mesmo nome.
Compartilhar	Um dos valores listados abaixo para a opção especificada.

Os significados e valores para as diversas opções são os seguintes:

Opção	Significado	Descrição
DDM_PKGRPLOPT(0x211C)	Sinalizador que especifica se um pacote existente com o mesmo ID da coleção e nome será substituído ou não.	DDM_PKGRPLALW (0x241F) Sim DDM_PKGRPLNA (0x2420) Não
DDM_STTDECDEL(0x2121)	O delimitador utilizado para o ponto decimal em instruções SQL no pacote.	DDM_DECDELPRD (0x243C) Ponto DDM_DECDELCMA (0x243D) Vírgula
DDM_STTSTRDEL(0x2120)	O delimitador utilizado para valores de cadeia em instruções SQL no pacote.	DDM_STRDELAP (0x2426) Apóstrofo DDM_STRDELQ (0x2427) Aspas Duplas
DDM_PKGISOLVL(0x2124)	O nível de isolamento para o pacote.	DDM_ISOLVLALL (0x2443) Tudo DDM_ISOLVLCHG (0x2441) Alterar DDM_ISOLVLCS (0x2442) Estabilidade do Cursor DDM_ISOLVLNC (0x2445) Sem Consolidação DDM_ISOLVLRR (0x2444) Leitura Repetível
DDM_PKGATHOPT(0x211E)	Sinalizador que especifica se as autorizações existentes no pacote serão mantidas ou não.	DDM_PKGATHKP (0x2425) Manter DDM_PKGATHRVK (0x2424) Revogar

DDM_QRYBLKCTL(0x2132)	O método a utilizar ao buscar linhas de dados para consultas no pacote.	DDM_FIXROWPRC (0x2418) Linha por vez DDM_LMTBLKPRC (0x2417) Bloco por vez
DDM_RDBRLSOPT(0x2129)	Quando liberar recursos do banco de dados adquiridos ao executar o pacote.	DDM_RDBRLSCMM (0x2438) Consolidação DDM_RDBRLSCNV (0x2439) Desalocação de Conversação
DDM_STTDATFMT(0x2122)	Formato para valores de dados recuperados.	DDM_ISODATFMT (0x2429) ISO DDM_USADATFMT (0x242A) EUA DDM_EURDATFMT (0x242B) Europeu DDM_JISDATFMT (0x242C) Padrão Industrial Japonês
DDM_STTTIMFMT(0x2123)	Formato para valores de hora recuperados.	DDM_ISOTIMFMT (0x242E) ISO DDM_USATIMFMT (0x242F) EUA DDM_EURTIMFMT (0x2430) Europeu DDM_JISTIMFMT (0x2431) Padrão Industrial Japonês

### Valor de Retorno

Zero se tiver êxito, diferente de zero se não tiver êxito. Se o valor de retorno for diferente de zero, você pode chamar `GetLastErrorString()` ou `GetLastErrorType()` para obter informações de erro adicionais.

### SetBindOwner()

`short SetBindOwner(BSTR NomeColeção, BSTR NomePacote, BSTR IDProprietário)`

### Descrição

Esta função permite especificar um proprietário diferente de seu ID de usuário para o pacote que você está ligando. Isto pode ser necessário se o seu ID de usuário não tiver as autorizações necessárias para ligar o pacote, mas o proprietário especificado as tiver.

### Parâmetros

Nome	Descrição
NomeColeção	O ID da coleção do pacote para o qual você deseja especificar o proprietário.

NomePacote	O nome do pacote para o qual você deseja especificar o proprietário.
Comentário	Uma cadeia que contém qualquer comentário que você queira salvar com a consulta. Se não houver comentário, passe este parâmetro como uma cadeia vazia ou NULL.
IDProprietário	O ID do proprietário desejado para o pacote que você está ligando.

### Valor de Retorno

Zero se tiver êxito, diferente de zero se não tiver êxito. Se o valor de retorno não for zero, você pode chamar `GetLastErrorString()` ou `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()` ou `GetLastSQLState()` para obter informações de erro adicionais.

## SetBusyWindowButton()

```
void SetBusyWindowButton(BSTR Texto)
```

### Descrição

Esta função especifica o texto exibido no botão Cancelar da janela ocupada.

### Parâmetros

Nome	Descrição
Texto	Uma cadeia que especifica o texto exibido no botão Cancelar da janela ocupada. O valor padrão é "Cancelar". Se você especificar uma cadeia vazia, o botão estará oculto. Independentemente do texto especificado, o botão sempre cancela, ou fecha, a janela.

### Valor de Retorno

Nenhum.

### Tópicos Relacionados

`SetBusyWindowMessage()`  
`SetBusyWindowMode()`  
`SetBusyWindowTitle()`  
`ShowBusyWindow()`

## SetBusyWindowMessage()

```
void SetBusyWindowMessage(BSTR Mensagem)
```

### Descrição

Esta função especifica o texto exibido na área de mensagens da janela ocupada.

## Parâmetros

Nome	Descrição
Mensagem	Uma cadeia que especifica o texto exibido na área de mensagens da janela ocupada.

## Valor de Retorno

Nenhum.

## Tópicos Relacionados

SetBusyWindowButton()

SetBusyWindowMode()

SetBusyWindowTitle()

ShowBusyWindow()

## SetBusyWindowMode()

void SetBusyWindowMode(short *Modo*)

## Descrição

Esta função determina se o QMF para Windows exibirá ou não a janela ocupada. A janela ocupada é útil para fornecer feedback ao usuário e para permitir que ele cancele uma ação pendente do banco de dados. Suas alterações entrarão em vigor na próxima vez que o QMF para Windows executar uma operação que cause a ocultação ou exibição da janela ocupada.

## Parâmetros

Nome	Descrição
Modo	Especifica quando o QMF para Windows exibe a janela ocupada:

Valor	Significado
0 (RSM_NEVER)	A janela não é exibida. Este é o padrão.
1 (RSM_WHENBUSY)	A janela é exibida quando o QMF para Windows está ocupado comunicando-se com o banco de dados. O QMF para Windows exibe automaticamente esta janela conforme apropriado.
2 (RSM_CLIENTCONTROLLED)	A janela é exibida após uma chamada de ShowBusyWindow(TRUE), e após uma chamada de ShowBusyWindow(FALSE). O cliente determina quando a janela é exibida.

### Valor de Retorno

Zero se tiver êxito, diferente de zero se não tiver êxito. Se o valor de retorno for diferente de zero, você pode chamar GetLastErrorString() ou GetLastErrorType() para obter informações de erro adicionais.

### Tópicos Relacionados

SetBusyWindowButton()  
SetBusyWindowMessage()  
SetBusyWindowTitle()  
SetParent()  
ShowBusyWindow()

## SetBusyWindowTitle()

void SetBusyWindowTitle(BSTR *Título*)

### Descrição

Esta função especifica o texto exibido na barra de título da janela ocupada.

### Parâmetros

Nome	Descrição
Título	Uma cadeia que especifica o texto exibido na barra de título da janela ocupada.

### Valor de Retorno

Nenhum.

### Tópicos Relacionados

SetBusyWindowButton()  
SetBusyWindowMode()  
SetBusyWindowMessage()  
ShowBusyWindow()

## SetGlobalVariable()

short SetGlobalVariable(BSTR *Nome*, BSTR *Valor*)

### Descrição

Esta função atribui um valor à variável global especificada. Este valor estará disponível para uso em consultas, formulários e procedimentos.

### Parâmetros

Nome	Descrição
Nome	Uma cadeia que contém o nome da variável que você deseja definir.

Valor	Uma cadeia que contém o valor que você deseja atribuir à variável especificada.
-------	---

### Valor de Retorno

Zero se tiver êxito, diferente de zero se não tiver êxito. Se o valor de retorno for diferente de zero, você pode chamar `GetLastErrorString()` ou `GetLastErrorType()` para obter informações de erro adicionais.

## SetHostVariable()

`short SetHostVariable(long IDConsulta, VARIANT Índice, VARIANT Valor)`

### Descrição

Esta função atribui um valor à variável do host especificada referenciada pela consulta. A consulta deve ser uma consulta estática referenciando variáveis de host (ou armazenadas com a consulta do QMF, ou criadas por `AddHostVariable()`). *Índice* pode especificar o índice numérico da variável do host ou o nome da variável do host.

### Parâmetros

Nome	Descrição
<i>IDConsulta</i>	O ID da consulta, conforme retornado de <code>InitializeStaticQuery()</code> .
<i>Índice</i>	Ou um número (tipo de variante <code>VT_I2</code> ) que especifica o índice da variável do host na consulta, ou uma cadeia (tipo de variante <code>VT_BSTR</code> ) que especifica o nome da variável do host.
<i>Valor</i>	O valor para a variável do host. Para especificar um valor nulo, o tipo da variante deve ser definido como <code>VT_EMPTY</code> .

### Valor de Retorno

Zero se tiver êxito, diferente de zero se não tiver êxito. Se o valor de retorno for diferente de zero, você pode chamar `GetLastErrorString()` ou `GetLastErrorType()` para obter informações de erro adicionais.

## SetOption()

`short SetOption(short Modo, VARIANT Valor)`

### Descrição

Esta função define o valor da opção especificado no QMF para Windows. Para algumas opções, as alterações não entram em vigor até que o QMF para Windows seja reinicializado. Sob condições normais, não reinicialize o QMF para Windows até ter destruído todas as instâncias do objeto da API do QMF para Windows.

## Parâmetros

Nome	Descrição
Opção	Especifica a opção a ser definida:

Valor	Significado
0 (RSO_SERVER_DEFINITION_FILE)	Nome do arquivo de definição do servidor.
1 (RSO_CPIC_DLL)	Nome do arquivo da DLL do Provedor de CPI-C.
2 (RSO_CPIC_TIMEOUT_WARNING)	Tempo limite de aviso do CPI-C (em segundos). Este limite não é utilizado pela API do QMF para Windows.
3 (RSO_CPIC_TIMEOUT_CANCEL)	Tempo limite de cancelamento do CPI-C (em segundos).
4 (RSO_TCP_TIMEOUT_WARNING)	Tempo limite de aviso do TCP (em segundos). Este limite não é utilizado pela API do QMF para Windows.
5 (RSO_TCP_TIMEOUT_CANCEL)	Tempo limite de cancelamento do TCP (em segundos).
6 (RSO_DISPLAY_NULLS_STRING)	A cadeia utilizada para exibir valores nulos.
7 (RSO_ENTER_NULLS_STRING)	A cadeia utilizada para inserir valores nulos.
8 (RSO_ENTER_DEFAULTS_STRING)	A cadeia utilizada para inserir valores padrão.
9 (RSO_TRACE_FILE_1)	Nome do arquivo de rastreamento 1.
10 (RSO_TRACE_FILE_2)	Nome do arquivo de rastreamento 2.
11 (RSO_TCP_TRACE_LEVEL)	Nível de rastreamento do TCP.
12 (RSO_CPIC_TRACE_LEVEL)	Nível de rastreamento do CPI-C.
13 (RSO_DDM_TRACE_LEVEL)	Nível de rastreamento do DDM.

Nome	Descrição
Valor	O valor com o qual a opção é definida.

### Valor de Retorno

Zero se tiver êxito, diferente de zero se não tiver êxito. Se o valor de retorno for diferente de zero, você pode chamar `GetLastErrorString()` ou `GetLastErrorType()` para obter informações de erro adicionais.

### Tópicos Relacionados

`GetOption()`

## SetParent()

short SetParent(long *JanelaPrinc*)

### Descrição

Esta função define a janela principal para diálogos. Normalmente, quando o QMF para Windows exibe um diálogo (na janela ocupada ou na caixa de diálogo Informações do Usuário), ele é centralizado e modal para a janela principal do QMF para Windows. Esta função permite forçar as caixas de diálogo do QMF para Windows para que sejam centralizadas e modais para a janela do aplicativo cliente.

### Parâmetros

Nome	Descrição
JanelaPrinc	O HWND da nova janela principal. Especifique NULL para utilizar a janela principal do QMF para Windows como pai.

### Valor de Retorno

Zero se tiver êxito, diferente de zero se não tiver êxito. Se o valor de retorno for diferente de zero, você pode chamar GetLastErrorString() ou GetLastErrorType() para obter informações de erro adicionais.

### Tópicos Relacionados

ShowBusyWindow()

## SetProcVariable()

short SetProcVariable(long *IDProc*, BSTR *Nome*, BSTR *Valor*)

### Descrição

Esta função atribui um valor à variável especificada. Este valor substitui a variável antes da execução do procedimento. Se o procedimento tiver uma ou mais variáveis, você deve chamar esta função para definir os valores das variáveis antes de chamar RunProc().

### Parâmetros

Nome	Descrição
IDProc	O ID do procedimento, conforme retornado de InitializeProc().
Nome	Uma cadeia que contém o nome da variável que você deseja definir.
Valor	Uma cadeia que contém o valor que você deseja atribuir à variável especificada.

### Valor de Retorno

Zero se tiver êxito, diferente de zero se não tiver êxito. Se o valor de retorno for diferente de zero, você pode chamar `GetLastErrorString()` ou `GetLastErrorType()` para obter informações de erro adicionais.

## SetVariable()

`short SetVariable(long IDConsulta, BSTR Nome, BSTR Valor)`

### Descrição

Esta função atribui um valor à variável especificada. Este valor substitui a variável antes da execução da instrução SQL. Se a instrução SQL tiver uma ou mais variáveis, você deve chamar esta função para definir os valores das variáveis antes de chamar `Open()` ou `Execute()`.

### Parâmetros

Nome	Descrição
IDConsulta	O ID da consulta, conforme retornado de <code>InitializeQuery()</code> .
Nome	Uma cadeia que contém o nome da variável que você deseja definir.
Valor	Uma cadeia que contém o valor que você deseja atribuir à variável especificada.

### Valor de Retorno

Zero se tiver êxito, diferente de zero se não tiver êxito. Se o valor de retorno for diferente de zero, você pode chamar `GetLastErrorString()` ou `GetLastErrorType()` para obter informações de erro adicionais.

## ShowBusyWindow()

`void ShowBusyWindow(BOOL Mostrar)`

### Descrição

Esta função instrui o QMF para Windows para que mostre ou oculte a janela ocupada. A janela ocupada é útil para fornecer feedback ao usuário e permite que ele cancele uma ação pendente do banco de dados. Esta função será efetiva apenas se você chamar `SetBusyWindowMode()` com um modo `RSM_CLIENTCONTROLLED`. Se você definir uma janela principal chamando `SetParent()`, a janela ocupada será modal para a janela especificada.

### Parâmetros

Nome	Descrição
Mostrar	Um valor diferente de zero mostra a janela ocupada; zero oculta a janela ocupada. Se for diferente de zero, a janela ocupada será exibida até que você chame <code>ShowBusyWindow()</code> com <i>Mostrar</i> definido como zero.

### Valor de Retorno

Nenhum.

## StartBind()

short StartBind(BSTR NomeColeção, BSTR NomePacote, BSTR TokenConsistência)

### Descrição

Esta função inicia o processo de ligação de um pacote no banco de dados.

### Parâmetros

Nome	Descrição
NomeColeção	O ID da coleção desejado para o pacote.
NomePacote	O nome desejado para o pacote.
TokenConsistência	Uma cadeia com 16 caracteres de comprimento contendo a representação hexadecimal de um token de oito bytes utilizado para assegurar a consistência entre o pacote ligado no banco de dados e um aplicativo que utiliza esse pacote. Quando uma seção for executada no pacote, você deverá fornecer este mesmo valor.

### Valor de Retorno

Zero se tiver êxito, diferente de zero se não tiver êxito. Se o valor de retorno não for zero, você pode chamar GetLastErrorString() ou GetLastErrorType(), GetLastErrorSQLCode(), GetLastErrorSQLError() ou GetLastErrorSQLState() para obter informações de erro adicionais.

### Tópicos Relacionados

EndBind()

CancelBind()



---

## Apêndice. Avisos

É possível que a IBM não ofereça, em outros países, os produtos, serviços ou recursos discutidos neste documento. Consulte o seu representante local IBM para obter informações sobre produtos e serviços atualmente disponíveis em sua área. Qualquer referência a um produto, programa ou serviço IBM não pretende declarar ou subentender que apenas aquele produto, programa ou serviço IBM possa ser utilizado. Qualquer produto, programa ou serviço funcionalmente equivalente, que não infrinja nenhum dos direitos de propriedade intelectual da IBM, pode ser utilizado em vez de um produto, programa ou serviço IBM. No entanto, a avaliação e verificação da operação de qualquer produto, programa ou serviço que não seja da IBM, é de responsabilidade do usuário.

A IBM poderá ter patentes ou pedidos de patentes pendentes, abrangendo o conteúdo deste documento. O fornecimento deste documento não lhe concede nenhuma licença sobre essas patentes. Você pode solicitar informações sobre licenças, por escrito, ao:

Gerente de Relações Comerciais e Industriais - IBM do Brasil  
Centro de Atendimento a Clientes IBM:  
CEP 22290-240  
Botafogo - RJ  
BRASIL.

**O parágrafo seguinte não se aplica ao Reino Unido ou a nenhum outro país onde tais cláusulas estejam inconsistentes com a lei local: A INTERNATIONAL BUSINESS MACHINES CORPORATION FORNECE ESTA PUBLICAÇÃO “ COMO ESTÁ ” SEM QUALQUER TIPO DE GARANTIA, EXPRESSA OU IMPLÍCITA, INCLUSIVE, PORÉM NÃO LIMITADA ÀS GARANTIAS IMPLÍCITAS DE NÃO-VIOLAÇÃO, NEGOCIABILIDADE OU ADEQUAÇÃO PARA UM DETERMINADO FIM. Alguns estados não aceitam a isenção de garantias expressas ou implícitas em algumas transações, conseqüentemente, é possível que esta declaração não se aplique a você.**

Estas informações podem conter imprecisões técnicas e erros tipográficos. Estas informações são alteradas periodicamente; as alterações será incorporadas em novas edições da publicação. A IBM pode fazer melhorias e/ou alterações no(s) produto(s) e/ou programa(s) descritos nesta publicação a qualquer hora, sem avisar.

Referências nesta publicação a sites Web que não são da IBM são fornecidas apenas para sua conveniência e de forma alguma servem como endosso

àquelas sites Web. Os materiais nestes sites Web não fazem parte dos materiais para este produto IBM e o uso destes fica por sua conta.

Os licenciados deste programa, que desejam obter informações sobre o mesmo, com a finalidade de possibilitar: (i) a troca de informações entre programas criados independentemente e outros programas (incluindo este) e (ii) o uso mútuo de informações que foram trocadas, devem entrar em contato com:

Gerência de Relações Comerciais e Industriais  
Av. Pasteur, 138-146 - Botafogo  
Rio de Janeiro, RJ  
CEP 22290-240

Tais informações poderão estar disponíveis, sujeitas aos termos e condições apropriadas, incluindo em alguns casos o pagamento de uma taxa.

O programa licenciado descrito nestas informações e todo material licenciado disponível para o mesmo são fornecidos pela IBM sob os termos do IBM Customer Agreement, IBM International Program License Agreement ou qualquer acordo equivalente entre nós.

Todos os dados de desempenho aqui contidos foram determinados em um ambiente controlado. Assim, os resultados obtidos em outros ambientes operacionais podem variar significativamente. Algumas medidas podem ter sido tomadas à nível de desenvolvimento de sistemas e não há garantia de que tais medidas serão iguais nos sistemas geralmente disponíveis. Além disso, algumas medidas podem ter sido estimadas através da extrapolação. Os resultados reais podem variar. Os usuários deste documento devem verificar os dados aplicáveis para seu ambiente específico.

Informações sobre os produtos que não são da IBM foram obtidas dos fornecedores destes produtos, de suas notificações publicadas ou de outras fontes disponíveis publicamente. A IBM não testou estes produtos e não pode confirmar a precisão do desempenho, compatibilidade ou qualquer outra reivindicação relacionada a produtos que não são da IBM. Dúvidas sobre a capacidade dos produtos que não são da IBM devem ser enviadas para os fornecedores destes produtos.

Todas as instruções relacionadas a tendências ou intenções futuras da IBM estão sujeitas a alteração ou remoção sem qualquer aviso, e representam apenas metas e objetivos.

Todos os preços da IBM apresentados, são preços de venda sugeridos pela IBM, são atuais e sujeitos a alteração sem aviso prévio. Os preços dos revendedores podem variar.

Estas informações servem apenas para fins de planejamento. As informações aqui contidas estão sujeitas à alterações antes dos produtos descritos tornarem-se disponíveis.

Estas informações contém exemplos de dados e relatórios utilizados em operações empresariais diárias. Para ilustrá-los o mais completamente possível, alguns exemplos incluem o nome de pessoas, empresas, marcas e produtos. Todos estes nomes são fictícios e qualquer semelhança com aqueles utilizados por uma empresa real terá sido mera coincidência.

#### LICENÇA DE DIREITOS AUTORAIS:

Estas informações contém programas de aplicação amostras na linguagem fonte, que ilustram técnicas de programação em várias plataformas operacionais. Estes programas amostra podem ser copiados, modificados e distribuídos em qualquer formato sem qualquer pagamento à IBM, para fins de desenvolvimento, uso, marketing ou distribuição de programas de aplicação adaptados à interface de programação de aplicação para a plataforma operacional para a qual os programas amostras foram desenvolvidos. Estes exemplos não foram completamente testados sob todas as condições. Conseqüentemente, a IBM não pode garantir ou inferir a confiabilidade, a durabilidade ou o funcionamento destes programas.

Se estas informações estiverem sendo visualizadas em cópia temporária, as fotos e ilustrações coloridas talvez não sejam exibidas.

---

## Marcas

Os termos a seguir são marcas da IBM Corporation nos Estados Unidos, em outros países ou em ambos:

ACF/VTAM	IBMLink
Advanced Peer-to-Peer	IMS
Networking	Language Environment
AIX	MVS/ESA
AIX/6000	MVS/XA
AS/400	OfficeVision/VM
CICS	OS/2
CICS/ESA	OS/390
CICS/MVS	PL/I
CICS/VSE	PROFS
COBOL/370	QMF
DATABASE 2	RACF
DataJoiner	S/390
DB2	SQL/DS
DB2 Universal Database	Virtual Machine/Enterprise
Distributed Relational	Systems Architecture
Database Architecture	Visual Basic
DRDA	VM/XA
DXT	VM/ESA
GDDM	VSE/ESA
IBM	VTAM

Java ou todas as marcas e logotipos baseados em Java, e Solaris são marcas da Sun Microsystems, Inc. nos Estados Unidos, em outros países ou em ambos.

Lotus e 1-2-3 são marcas da Lotus Development Corporation nos Estados Unidos, outros países ou em ambos.

Microsoft, Windows e Windows NT são marcas ou marcas registradas da Microsoft Corporation.

Outros nomes de serviços, produtos ou empresas, indicados por um duplo asterisco (\*\*), podem ser marcas ou de serviços de terceiros.

---

# Índice Remissivo

## A

### abertura

- arquivos de consulta
  - orientada 24
- arquivos SQL salvos 13
- consultas orientadas no servidor de banco de dados 24
- consultas SQL no servidor de banco de dados 14
- formulários salvos no servidor de banco de dados 38
- procedimentos no banco de dados 43

AddDecimalHostVariable() 75

AddHostVariable() 76

administração 4

agrupamento de resultados da consulta 29, 51

alteração de linha

editor de tabela 60

alteração de senhas 3

aplicações de exemplos 66

arquivos

exportação de dados para 63

arquivos de job, criação 49

Avisos 149

## B

banco de dados

segurança 2

barra de ferramentas

adição de botões 6

movimentação de botões 6

personalização 6

remoção de botões 7

BindDecimalHostVariable() 77

BindHostVariable() 77

BindSection() 78

Bloqueio de chamadas 74

botões de ação

consulta orientada 17

## C

cadeias de conta 4

cálculos

formulários 33

CancelBind() 79

ChangePassword() 79

classificação de resultados da

consulta 28, 50

ClearList() 80

Close() 80

colunas

formulários 33

inclusão em consultas

orientadas 19

colunas, reordenação 28, 50

colunas, seleção 27

Commit() 81

complemento

Excel 66

CompleteQuery() 81

condições

formulários 33

condições de junção

criação em consultas

orientadas 22

condições de linha

contendo 21

entre 21

igual a 20

inclusão 21

inicializar com 21

maior que 21

maior que ou igual a 21

menor ou igual a 21

menor que 21

nulo 21

terminar com 21

utilizando 20

condições de ordenação

inclusão 20

utilizando 20

Conexão ao banco de dados 74

consulta orientada

botões de ação 17

consultas

construção complexa 19

consultas complexas

construção 19

consultas estáticas

criando 55

execução 57

uso de variáveis de

substituição 55

consultas orientadas

abertura de arquivos salvos 24

conversão em SQL 23

consultas orientadas (*continuação*)

criação de condições de

junção 22

criando 17

execução 18

exibição de SQL 23

inclusão de colunas 19

inclusão de tabelas 18

salvando no servidor de banco de dados 24

salvar como arquivos 23

uso de múltiplas tabelas 22

uso de variáveis de

substituição 23

uso do SQL em 22

Consultas SQL

abertura de arquivos salvos 13

abertura e um documento

novo 9

execução no servidor de banco de dados 9

impressão 15

salvando no servidor de banco de dados 14

salvar em arquivos 13

visualização de impressão 15

contendo (condição de linha) 21

conversão da formatação em um formulário 29, 51

CopyToClipboard() 82

Correio da Internet 65

criação de arquivos de job 49

criação de um procedimento

linear 41

criando

consultas estáticas 55

## D

DB2 Forms 61

DeleteQMFObject() 83

desenhar objeto 46

detalhes

formulários 33

## E

É (operador de condição de

linha) 21

edição de objeto 46

editor de tabela 59

alteração de linha 60

- editor de tabela 59 (*continuação*)
  - inclusão de linha 60
  - pesquisa de linhas 59
- EndBind() 83
- entre (condição de linha) 21
- Enviar Para 65
- Excel
  - complemento 66
- execução
  - consulta SQL em um servidor de banco de dados 9
  - consultas estáticas 57
  - consultas orientadas 18
- execução de objeto 46
- Execute() 84
- ExecuteEx() 84
- ExecuteStoredProcedure() 85
- ExecuteStoredProcedureEx() 86
- exibição
  - resultados 9
  - SQL 9
  - SQL em consultas orientadas 23
- exibição de objetos 46
- exibição de resultados 9
- Export() 88
- exportação de dados
  - em um servidor de banco de dados 65
  - para arquivos 63
  - para outras tabelas 65
- exportando
  - relatórios 39
- ExportForm() 90
- ExportReport() 91

**F**

- FastSaveData() 92
- FetchNextRow() 93
- FetchNextRowEx() 94
- FetchNextRows() 95
- FetchNextRowsEx() 97
- final
  - formulários 33
- FlushQMFCache() 97
- fontes
  - exibição da consulta 10
  - exibição de resultados 29, 50
- formatação de resultados de consulta
  - numéricos 29, 51
- formular consultas
  - criando 11
- formulário
  - principal 33
- formulários
  - abertura de um arquivo salvo 37, 42

- formulários (*continuação*)
  - cálculos 33
  - colunas 33
  - condições 33
  - detalhes 33
  - final 33
  - HTML 34
  - opções 34
  - página 34
  - produção de um relatório 34
  - quebras 33
  - salvando no servidor de banco de dados 37, 42
  - salvar em arquivos 37, 42

**G**

- GetColumnCount() 98
- GetColumnDataValue() 98
- GetColumnHeader() 98
- GetColumnHeaderEx() 99
- GetColumnHeadings() 100
- GetColumnValue() 101
- GetColumnValueEx() 101
- GetDefaultServerName() 102
- GetGlobalVariable() 102
- GetHostVariableNames() 102
- GetHostVariableTypes() 103
- GetLastErrorString() 103
- GetLastErrorType() 104
- GetLastSQLCode() 105
- GetLastSQLError() 106
- GetLastSQLState() 107
- GetOption() 107
- GetOptionEx() 109
- GetProcText() 109
- GetProcVariables() 110
- GetQMFOBJECTInfo() 111
- GetQMFOBJECTInfoEx() 113
- GetQMFOBJECTList() 114
- GetQMFOBJECTListEx() 115
- GetQMFPProcText() 116
- GetQMFFQueryText() 117
- GetQueryText() 117
- GetQueryVerb() 118
- GetResourceLimit() 118
- GetResourceLimitEx() 122
- GetRowCount() 123
- GetServerList() 123
- GetServerListEx() 124
- GetStoredProcedureResultSets() 125
- GetVariables() 126
- GetVariablesEx() 126

**H**

- HTML
  - formulários 34

**I**

- igual a (condição de linha) 20
- impressão
  - Consultas SQL 15
  - procedimento 44
  - relatórios 38
- impressão de resultados da consulta 31, 52
- inclusão
  - condições de linha 21
  - condições de ordenação 20
- inclusão de linha
  - editor de tabela 60
- inclusão de objetos em listas 46
- inicializar com (condição de linha) 21
- início de sessão 2
- InitializeProc() 127
- InitializeQuery() 128
- InitializeServer() 128
- InitializeStaticQuery() 130
- IsStatic() 130

**L**

- linhas, seleção 27
- listando
  - objetos 45
- listas
  - abertura de arquivos salvos 47
- listas, inclusão de objetos 46
- listas, remoção de objetos 47

**M**

- maior que (condição de linha) 21
- maior que ou igual a (condição de linha) 21
- menor que (condição de linha) 21
- menor que ou igual a (condição de linha) 21
- múltiplas consultas
  - exibição simultânea 10
- múltiplos documentos de consulta 10

**N**

- Não é (operador de condição de linha) 21
- nova
  - consulta SQL 9
  - consultas orientadas 17
  - formular consultas 11
- nulo (condição de linha) 21

**O**

- objetos
  - listando 45

- opções
  - formulários 34
- Open() 130
- operador de condição de linha
  - É 21
  - Não é 21

## P

- página
  - formulários 34
- pesquisa
  - editor de tabela 59
- Prepare() 131
- principal
  - formulários 33
- PrintReport() 132
- procedimento
  - impressão 44
- procedimento com lógica 41
- procedimento REXX 41

## Q

- quebras
  - formulários 33

## R

- redimensionamento de colunas e linhas 27
- Referência da API 75
- ReinitializeServer() 132
- relatórios
  - exportando 39
  - impressão 38
  - produção de um relatório usando formulários 34
  - visualização 34
- remoção de objetos de listas 47
- reordenação de colunas 28, 50
- resultados da consulta,
  - agrupamento 51
- resultados da consulta,
  - classificação 28, 50
- resultados da consulta,
  - impressão 31, 52
- resultados da consulta, resumo 30, 51
- resultados da consulta,
  - salvamento 30, 52
- resultados de consulta,
  - agrupamento 29
- resultados de consulta,
  - formatação 29, 51
- resultados de consulta, salvamento em arquivos. 30, 52
- resultados de consulta,
  - visualização 30, 52

- resumo de resultados da consulta 30, 51
- Rollback() 133
- RunProc() 133

## S

- salvamento de resultados da consulta 30, 52
- salvamento de resultados da consulta em arquivos 30, 52
- salvando
  - consultas orientadas como arquivos 23
  - consultas orientadas no servidor de banco de dados 24
  - consultas SQL em arquivos 13
  - consultas SQL no servidor de banco de dados 14
  - formulários em arquivos 37, 42
  - formulários no servidor de banco de dados 37, 42
- SaveData() 134
- SaveQMFProc() 136
- SaveQMFQuery() 137
- Seleção de colunas e linhas 27
- senhas
  - correção 3
- servidor
  - definição 1
- servidor do banco de dados
  - exportação de dados para 65
- SetBindOption() 137
- SetBindOwner() 139
- SetBusyWindowButton() 140
- SetBusyWindowMessage() 140
- SetBusyWindowMode() 141
- SetBusyWindowTitle() 142
- SetGlobalVariable() 142
- SetHostVariable() 143
- SetOption() 143
- SetParent() 145
- SetProcVariable() 145
- SetVariable() 146
- ShowBusyWindow() 146
- SQL
  - uso em consultas orientadas 22
- StartBind() 147

## T

- tabelas
  - exportação de dados para 65
  - inclusão em consultas orientadas 18
- tabelas múltiplas
  - em consultas orientadas 22
- terminar com 21

## V

- variáveis de substituição
  - em consultas SQL 12
  - execução de consultas SQL com 12
  - substituição com variáveis do host 55
  - uso em consultas estáticas 55
  - uso em consultas orientadas 23
- variáveis do host
  - uso em consultas estáticas 55
- visualização
  - consultas impressas 15
  - procedimento impresso 43
  - relatórios 34
- visualização de impressão
  - consultas orientadas 25
- visualização de resultados da consulta 30, 52



---

# Comentários do Leitor

Query Management Facility  
Informações Iniciais do QMF para Windows  
Versão 7

Publicação N° S517-7008-00

Neste formulário, faça-nos saber sua opinião sobre este manual. Utilize-o se encontrar algum erro, ou se quiser externar qualquer opinião a respeito (tal como organização, assunto, aparência...) ou fazer sugestões para melhorá-lo.

Para pedir publicações extras, fazer perguntas ou tecer comentários sobre as funções de produtos ou sistemas IBM, fale com o seu representante IBM.

Quando você envia seus comentários, concede direitos, não exclusivos, à IBM para usá-los ou distribuí-los da maneira que achar conveniente, sem que isso implique em qualquer compromisso ou obrigação para com você.

Não se esqueça de preencher seu nome e seu endereço abaixo, se deseja resposta.

Comentários:

---

Nome

---

Endereço

---

Companhia ou Empresa

---

Telefone



Centro Industrial IBM Brasil  
Centro de Traduções  
Caixa Postal 71  
13001-970 Campinas, SP  
BRASIL.





Número de Arquivo:

Número do Programa: 5675-DB2  
5697-F42  
5697-G24  
5697-G22  
5648-D35  
5697-G23

Impresso na Dinamarca

S517-7008-00



Spine information:



QMF

Informações Iniciais do QMF para Windows

Versão 7