

ÍNDICE

| | |
|--|-----------|
| INTRODUÇÃO AO MS-ACCESS: | 1 |
| RECAPITULAÇÃO:..... | 1 |
| <i>Tabelas:</i> | 1 |
| <i>Consultas:</i> | 1 |
| <i>Formulários:</i> | 1 |
| <i>Relatórios:</i> | 1 |
| SEGURANÇA: | 2 |
| ANTES DE COMEÇAR : | 2 |
| PROTEÇÃO DOS DADOS | 2 |
| CONSULTAS: | 4 |
| USANDO SQL NO MS-ACCESS..... | 4 |
| CONSULTAS UNIÃO | 4 |
| CONSULTA DEFINIÇÃO DE DADOS..... | 6 |
| <i>Tipos de dados do SQL compatíveis com o Access</i> | 6 |
| CONSULTA PASSO ATRAVÉS | 8 |
| CRIAR SUB-CONSULTAS..... | 8 |
| DIFERENÇAS ENTRE CONSULTAS AÇÃO E CONSULTAS SELEÇÃO | 8 |
| ALTERANDO DADOS COM CONSULTAS AÇÃO | 9 |
| CONSULTA CRIAÇÃO DE TABELA: | 9 |
| CONSULTA EXCLUSÃO:..... | 10 |
| CONSULTA ATUALIZAÇÃO: | 10 |
| CONSULTA ANEXAÇÃO:..... | 11 |
| ESTRUTURANDO FORMULÁRIOS | 12 |
| CRIANDO CONTROLES..... | 12 |
| <i>Controles vinculados, não vinculados e calculados</i> | 12 |
| <i>Definindo propriedades de controle</i> | 13 |
| PERSONALIZANDO FORMULÁRIOS | 14 |
| <i>Alterando ordem de tabulação</i> | 14 |
| <i>Criando outros controles utilizando a caixa de ferramentas</i> | 15 |
| <i>Fornecendo uma lista de opções com uma caixa de listagem ou uma caixa de combinação</i> | 16 |
| CRIANDO UM FORMULÁRIO/SUB-FORMULÁRIO SEM A AJUDA DO ASSISTENTE..... | 17 |
| <i>Vinculando um formulário e um sub-formulário</i> | 19 |
| CALCULANDO UM TOTAL PARA UM GRUPO DE REGISTROS..... | 19 |
| ESTRUTURANDO RELATÓRIOS | 21 |
| ADICIONANDO SEÇÕES | 21 |
| <i>Cabeçalho do Relatório</i> | 21 |
| <i>Cabeçalho da Página</i> | 21 |
| <i>Para adicionar cabeçalho e rodapé a um relatório ou página</i> | 21 |
| <i>Cabeçalho do Grupo</i> | 21 |
| <i>Para adicionar cabeçalho e rodapé a um grupo</i> | 21 |
| <i>Seção detalhe</i> | 23 |
| <i>Rodapé do grupo</i> | 23 |
| <i>Rodapé da página</i> | 23 |
| <i>Rodapé do relatório</i> | 23 |

| | |
|--|-----------|
| USANDO EXPRESSÕES EM RELATÓRIOS | 24 |
| MACROS | 25 |
| O QUE É UMA MACRO? | 25 |
| PORQUE USAR MACROS? | 25 |
| <i>Fazer seus formulários e relatórios trabalhem juntos</i> | <i>25</i> |
| <i>Encontrar e filtrar registros automaticamente.....</i> | <i>26</i> |
| <i>Definir valor em controles.....</i> | <i>26</i> |
| <i>Garantir a precisão dos dados.....</i> | <i>26</i> |
| <i>Definir propriedade de formulários, relatórios e controles.....</i> | <i>26</i> |
| <i>Automatizar transferências de dados</i> | <i>26</i> |
| <i>Criar o seu próprio ambiente de trabalho personalizado</i> | <i>26</i> |
| CRIANDO UMA MACRO | 27 |
| <i>Para criar uma macro</i> | <i>27</i> |
| <i>A Janela Macro</i> | <i>27</i> |
| <i>Adicionando ações a uma Macro</i> | <i>28</i> |
| <i>Definindo argumentos</i> | <i>29</i> |
| <i>Salvando uma Macro.....</i> | <i>29</i> |
| <i>Criando um grupo de macro</i> | <i>30</i> |
| <i>Usando condições em macros</i> | <i>32</i> |
| LISTA DE AÇÕES DE MACRO | 34 |
| RESPONDENDO A EVENTOS EM FORMULÁRIOS..... | 36 |
| <i>Fazendo Formulários trabalharem em conjunto.....</i> | <i>36</i> |
| <i>Atribuindo macros de validação a eventos de formulário e de controle.....</i> | <i>37</i> |
| <i>Exibindo mensagens de validação diferentes para condições diferentes.....</i> | <i>38</i> |
| CRIANDO NOSSOS PRÓPRIOS MENUS | 41 |
| <i>Definição do Menu</i> | <i>41</i> |
| <i>Definição da Barra de Menu.....</i> | <i>42</i> |
| <i>Atribuindo uma Barra de Menu a um formulário</i> | <i>42</i> |
| MÓDULOS | 43 |
| ESTRUTURA DO MÓDULO | 43 |
| DECLARANDO VARIÁVEIS | 43 |
| <i>Tipos de Dados.....</i> | <i>44</i> |
| USANDO DADOS CONTIDOS NOS FORMULÁRIOS | 45 |
| OBJETOS DE BANCO DE DADOS..... | 46 |
| VALIDAÇÃO DE CAMPOS COM MÓDULOS | 47 |

Introdução ao MS-Access:

Recapitulação:

Tabelas:

Uma coleção de dados de mesmo assunto ou tópico, uma tabela armazena os dados em Registros (linhas) e Campos (colunas)

Consultas:

Objeto que seleciona registros e campos armazenados nas tabelas. Você pode pensar numa consulta como uma pesquisa num conjunto particular de dados.

Formulários:

Objeto no qual você coloca controles para a entrada, exibição e edição de dados nos campos.

Relatórios:

Objeto que apresenta os dados formatados e organizados segundo as especificações do usuário.

Segurança:

Antes de começar :

Deve-se criar uma cópia de reserva do banco de dados do grupo de trabalho, SYSTEM.MDA, onde o Microsoft Access armazena as especificações das opções de cada usuário e as configurações da barra de ferramentas. Em um ambiente de multi-usuário, o banco de dados do grupo de trabalho também contém as informações de cadastro de segurança do seu grupo. Se o seu arquivo SYSTEM.MDA for perdido ou danificado, você não poderá iniciar o Microsoft Access. Uma vez feita uma cópia de reserva desse arquivo, só será preciso copiá-lo novamente caso haja mudanças nas opções, barra de ferramentas ou nas informações da cadastro de segurança.

Proteção dos dados

Proteger uma informação pode ser uma parte importante no desenvolvimento de um banco de dados do Microsoft Access. Através da proteção ao seu banco de dados, você poderá controlar o que um usuário ou um grupo de usuários pode fazer com objetos do banco de dados.

A segurança do Microsoft Access é composta de várias partes que precisam estar no lugar antes que a segurança se conclua: grupos de trabalho, usuário e contas do grupo, posse, e atribuições de permissão.

Para proteger um banco de dados:

| | |
|---|--|
| 1 | Atribuir senha ao usuário Administrador ; sair do Access e Logar como Administrador |
| 2 | Definir grupos de usuários |
| 3 | Definir permissões do grupo |
| 4 | Criar usuários |
| 5 | Tirar as permissões dos grupos Usuários e Administradores |
| 6 | Colocar os usuários que vão administrar o banco de dados no grupo Administradores |

Importante: Você precisa remover as permissões do usuário administrador e do grupo de usuários, que inclui todos os usuários em um grupo de trabalho. Até que você remova estas permissões, os usuários poderão ter permissões involuntárias para objetos. Tendo, uma vez, as permissões removidas, os usuários passarão a ter somente as que você atribuiu.

Para assegurar a proteção do banco de dados, é importante acrescentar uma senha à conta do administrador e às contas do proprietário e administrador designadas. Adicionalmente, você poderá acrescentar uma senha às contas criadas para usuários ou instruí-los para acrescentar às suas próprias senhas.

Observações: A segurança depende da relação entre contas do grupo de trabalho do Microsoft Access e das atribuições de permissão em seu banco de dados. O Microsoft Access armazena informação de conta com o grupo de trabalho e atribuições de permissões no arquivo (.MDB) do banco de dados.

Exercício:

Implemente a segurança no banco de dados do curso, criando três grupos:

- DBA - administrador (Todas as permissões)
- Gerentes - usuário (somente para leitura)
- Digitadores - usuário (com permissões para leitura, gravação, exclusão e atualização)

Consultas:

Usando SQL no MS-Access

A Structured Query Language SQL é uma linguagem muito usada na consulta, atualização e gerenciamento de banco de dados relacionais. Cada consulta criada no MS-Access tem uma instrução SQL associada que define as ações daquela consulta. Normalmente é mais conveniente usar a grade QBE e as ferramentas gráficas de estruturação de consulta do Access para criar e modificar consultas. Mas se você está familiarizado com SQL, pode usar instruções SQL para:

Consultas União

É uma consulta específica SQL que cria um resultado contendo dados de todos os registros especificados em duas ou mais tabelas. Pelo contrário, uma consulta baseada em uma associação cria um dynaset contendo dados somente de registros cujos campos relacionados encontram uma condição especificada.

Por exemplo, uma consulta união da tabela clientes e data da tabela fornecedores resulta um *Resultado* que contém todos os registros especificados em ambas as tabelas, a de clientes e a de fornecedores.

A consulta união a seguir, consiste de duas instruções SELECT que retornam os nomes de empresa e cidades que estejam tanto na tabela Fornecedores como na tabela Clientes, e sejam do Brasil.

A seguinte consulta união consiste de duas instruções SQL SELECT que retornam os nomes de companhias e cidades que estão nas tabelas Fornecedores e Clientes e são do Brasil.

```
SELECT [Nome da empresa], [Cidade]
FROM [Fornecedores]
WHERE Pais="Brasil"
```

```
UNION SELECT [Nome da empresa], [Cidade]
FROM [Clientes]
WHERE Pais="Brasil";
```

MS-Access 2.0 Avançado

A seguinte consulta união seleciona o nome de todas as empresas e cidades que estejam nas tabelas Fornecedores e Clientes e classifica alfabeticamente os dados por cidade.

```
SELECT [Nome da empresa], [Cidade]  
FROM [Fornecedores]
```

```
UNION SELECT [Nome da empresa], [Cidade]  
FROM [Clientes]  
ORDER BY [Fornecedores], [Cidade];
```

A consulta união a seguir renomeia o campo "Nome da empresa" para "Nome do cliente/fornecedor" na saída da consulta;

```
SELECT [Nome da empresa] AS [Nome do cliente/fornecedor], [Cidade]  
FROM [Fornecedores]  
UNION SELECT [Nome da empresa] AS [Nome do cliente/fornecedor], [Cidade]  
FROM [Clientes];
```

A consulta união a seguir utiliza a instrução UNION ALL para recuperar registros duplicados:

```
SELECT [Nome da empresa], [Cidade]  
FROM [Fornecedores]  
  
UNION ALL SELECT [Nome da empresa], [Cidade]  
FROM [Clientes];
```

Consulta Definição de Dados

Este tipo de consulta cria uma tabela sem adicionar nenhum registro a ela, é interessante quando precisamos criar tabelas temporárias. Veja o exemplo:

```
CREATE TABLE [SÓCIOS]
(Código COUNTER,
Nome TEXT (50),
Rua TEXT (50),
(CONSTRAINT CódigoDoSócio PRIMARY KEY ([Código]));
```

No exemplo acima é criada uma tabela Sócios com os seguintes campos:

| Campo | Tipo | Tamanho |
|--------|----------|---------|
| Código | Contador | 4 bytes |
| Nome | Texto | 50 |
| Rua | Texto | 50 |

A palavra chave **CONSTRAINT** indica que em seguida vem o nome de um índice chamado CódigoDoSócio, entre parêntesis vem o nome do campo que faz parte deste índice e opcionalmente a cláusula **PRIMARY KEY** para indicar a chave primária da tabela.

Tipos de dados do SQL compatíveis com o Access

| Tipo de dado | Tamanho | Descrição |
|--------------|---------|--|
| BINARY | 1 byte | Para consultas em tabelas anexadas em produtos de banco de dados que definem um tipo de dados Binário. O Microsoft Access não usa o tipo de dados Binário. |
| BIT | 1 byte | Valores Sim e Não e campos que contém apenas um entre dois valores. |
| BYTE | 1 byte | Um valor inteiro entre 0 e 255. |
| COUNTER | 4 bytes | Um número incrementado automaticamente pelo Microsoft Access quando um novo registro é adicionado a uma tabela. No Jet database engine, o tipo de dado para este valor é Long. |
| CURRENCY | 8 bytes | Um inteiro dimensionado entre -922.337.203.685.477,5808 e 922.337.203.685.477,5807. |

| Tipo de dado | Tamanho | Descrição |
|---------------------------|---------------------|--|
| DATETIME(Consulte DOUBLE) | 8 bytes | Um valor data ou hora entre os anos 100 e 9999. |
| SINGLE | 4 bytes | Um valor ponto flutuante de precisão simples com um intervalo de -3,402823E38 a -1,401298E-45 para valores negativos, 1,401298E-45 a 3,402823E38 para valores positivos e 0. |
| DOUBLE | 8 bytes | Um valor ponto flutuante de precisão dupla com um intervalo de -1,79769313486232E308 a -4,94065645841247E-324 para valores negativos, 4,94065645841247E-324 a 1,79769313486232E308 para valores positivos e 0. |
| SHORT | 2 bytes | Um inteiro curto entre -32.768 e 32.767 |
| LONG | 4 bytes | Um inteiro longo entre -2.147.483.648 e 2.147.483.647. |
| LONGTEXT | 1 byte por caracter | Zero até um máximo de 1,2 gigabytes. (Consulte Text.) |
| LONGBINARY | Como definido | Zero a um máximo de aproximadamente 1 gigabyte. Usado para objetos OLE. |
| TEXT | 1 byte por caracter | Zero a 255 caracteres. |

Exercício

Criar uma consulta Definição de Dados para criar uma tabela com a seguinte estrutura:

| Nome do Campo | Tipo do dado | Tamanho |
|------------------------|--------------|---------|
| Código do Produto | Contador | |
| Nome do Produto | Texto | 40 |
| Quantidade Por Unidade | Texto | 20 |
| Preço Unitário | Moeda | |
| Unidades em Estoque | Número | Inteiro |
| Categoria | Texto | 15 |
| Fornecedor | Texto | 40 |

O nome da tabela a ser criada será *Informações dos produtos*. Como chave primária será utilizado o campo *Código do Produto*.

Consulta Passo Através

As consultas passo através não são executadas pelo Access, elas são executadas pelo servidor SQL, por isso não é possível usar o SQL do Access, visto que ele não segue o padrão ANSI como os servidores SQL.

Esta consulta manipula os dados diretamente no servidor então as tabelas utilizadas não precisam estar anexadas.

Criar sub-consultas

para criar uma sub-consulta, você usa a grade QBE, mas insere uma instrução SQL SELECT como critério.

Para maiores informações consulte *Criando uma Sub-consulta* dentro de *SQL: sub-consultas* na ajuda do MS-Access.





Diferenças entre consultas Ação e consultas seleção

Quando se cria uma consulta, o MS-Access cria uma consulta seleção, a não ser que se escolha um tipo diferente no menu **consulta**.

Quando uma consulta seleção é executada, o MS-Access exibe o *Dynaset*, os registros recuperados de suas tabelas. Pode-se, então, visualizar o dynaset e fazer alterações em seus dados. Entretanto, ao fazer alterações, elas são inseridas registro a registro.

Caso haja um grande número de alterações similares, você pode poupar tempo usando uma *consulta ação*. Uma consulta ação faz quaisquer alterações em apenas uma operação. Por exemplo, caso deseje excluir todos os produtos que tenham sido descontinuados, você pode executar uma consulta ação que remova todos esses produtos.

Para se garantir que alterações indesejadas não sejam feitas se você executar **inadvertidamente** uma consulta ação, o MS-Access exibe na janela Banco de dados, um ícone especial para identificar as quatro consultas ação.

-  Consulta Anexação
-  Consulta Atualização
-  Consulta Criação de Tabela
-  Consulta Exclusão

MS-Access 2.0 Avançado

Alterando dados com consultas Ação

Consultas Ação:

Com as consultas ação, você pode criar novas tabelas ou alterar os dados de tabelas já existentes. Todas as **Consultas Ação** são feitas com base numa **consulta seleção**, então para se criar qualquer tipo de consulta ação, devemos criar as consultas seleção que nos trarão os dados que desejamos manipular pelas consultas ação, podendo assim atualizar, excluir, exportar e anexar dados.

O MS-Access proporciona quatro tipos de consultas ação.

Consulta criação de tabela:

Cria uma nova tabela a partir de outra ou parte de outra tabela com um novo nome definido pelo usuário e adicionando os registros especificados pelo critério.

Consulta exclusão:

Exclui os registros de uma ou mais tabelas que atendem a um critério.

Consulta anexação:

Adiciona uma grupo de registros de uma ou mais Tabelas para outra ou mais tabelas.

Consulta atualização:

Altera ou dados em um grupo de registros de uma ou mais tabelas.

Consulta Criação de Tabela:



1. Após criarmos a consulta seleção com os campos desejados, clique no botão da consulta **criação de tabela**, em seguida forneça o nome da tabela a ser criada. Se for necessário, você pode criar esta tabela em outro banco de dados, clicando no botão **outro banco de dados** e forneça o seu nome. Depois pressione o botão Ok.
2. Quando você clicar no botão **Executar**, o MS-Access informa quantas linhas (registros) serão incluídos na nova tabela, nesta mensagem, você pode confirmar a criação ou cancelar a inclusão.
3. Se você clicar no botão **Modo Folha de Dados**, a consulta seleção será executada e você verá os registros que vão para a nova tabela.

Exercício I

Crie uma consulta de Criação de tabela para listar o *Nome do produto*, *Quantidade em estoque* e o *Fornecedor*, *Código do Fornecedor*, onde a quantidade em estoque seja menor que 20. O nome da tabela de saída será *Produtos em Baixa*.

Salve a consulta como *Cria Produtos em Baixa*.



Consulta exclusão:

1. Quando desejamos excluir vários dados de uma tabela com dados alguns dados em comum, nós devemos criar uma consulta seleção para exibir estes registros e depois transformará numa consulta exclusão.
2. Quando você clicar no botão *Executar*, o MS-Access informa quantas linhas (registros) serão excluídas da tabela, nesta mensagem, você pode confirmar a eliminação ou não.
3. Se você clicar no botão *Modo Folha de Dados*, a consulta seleção será executada e você verá os registros que vão excluídos.

Exercício II

Crie uma consulta para excluir todos da tabela de produtos, todos os produtos que estejam com a quantidade em estoque abaixo de 20.

Salve com o nome de Exclui Produtos em Baixa.



Consulta Atualização:

1. A consulta atualização atualiza os dados de um ou mais campos, cria-se a consulta seleção com os campos desejado e depois pressione o botão da *consulta atualização* e na linha *atualizar para* você coloca o valor ou expressão que irá atualizar o campo.

Exercício III

Crie uma consulta para atualizar o campo Unidades em estoque em 50% se o valor for diferente de 0 (zero) ou 50 se for igual a 0 (zero).

Salve com o nome *Atualiza Produtos*.

Consulta anexação:

1. A consulta anexação nos permite anexar dados de uma tabela a outra. Para isso devemos criar uma consulta seleção com os campos que desejamos anexar, e depois clicamos no botão da consulta anexação e está pronta a consulta anexação.
2. Quando você clicar no botão **Executar**, o MS-Access informa quantas linhas (registros) serão anexadas da tabela, nesta mensagem, você pode confirmar a anexação ou não.
3. Se você clicar no botão **Modo Folha de Dados**, a consulta seleção será executada e você verá os registros que vão ser anexados.

Exercício IV

Criar uma consulta para Anexar os campos para a tabela **Informações do produto**.

Estruturando formulários

Criando Controles

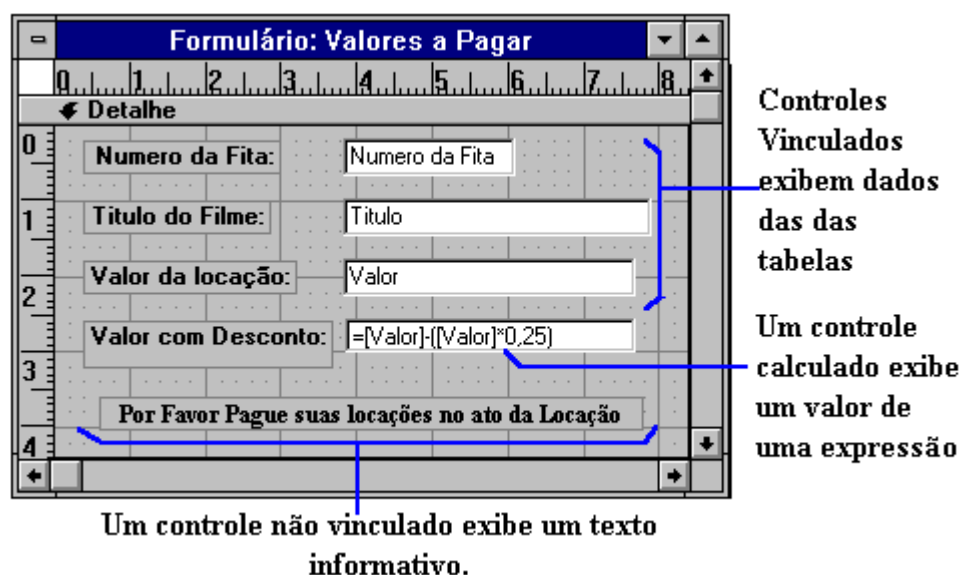
Quando começa um formulário em branco, você mesmo adiciona todos os controles. Se criou um formulário com o assistente de formulário, pode-se também desejar adicionar controles para exibir dados adicionais ou para tornar o formulário mais atraente ou mais fácil de usar. Por exemplo, você pode desejar adicionar uma etiqueta que inclua instruções sobre como utilizar um formulário no *modo formulário*.

Controles vinculados, não vinculados e calculados

Quando se adiciona um controle a um formulário, determina-se de onde ele obtém a informação que exibe. Por exemplo, você pode adicionar um controle caixa de texto que exibe um produto da companhia a partir do campo NomeDoProduto da tabela produtos. A caixa de texto obtém os valores a partir do campo Nome do Produto por ter sido conectado ou vinculado àquele campo.

Os controles também podem exibir informações que não estejam armazenadas no banco de dados, tais como instruções, legendas e valores calculados. Conhecer a origem dos dados de um controle ajuda a decidir que tipo de controle usar e como criá-lo.

Os controles podem ser vinculados, não vinculados e calculados. A ilustração seguinte mostra os três tipos de controle.



Se você comparar um formulário no modo estrutura e no modo folha de dados, poderá ver a conexão entre os controles do formulário e os campos da tabela de origem do formulário.

Definindo propriedades de controle

No MS-Access, as propriedades são usadas para determinar as características das tabelas, consultas, campos, formulários e relatórios. Cada controle de um formulário tem também propriedades. As propriedades de controle determinam a estrutura e a aparência de um controle bem como as características do texto ou dos dados que ele contém. Por exemplo, você pode definir propriedades que formatem os valores de um controle como moeda e os exiba em negrito.

Pode-se ver as propriedades de um controle exibindo sua folha de propriedade.

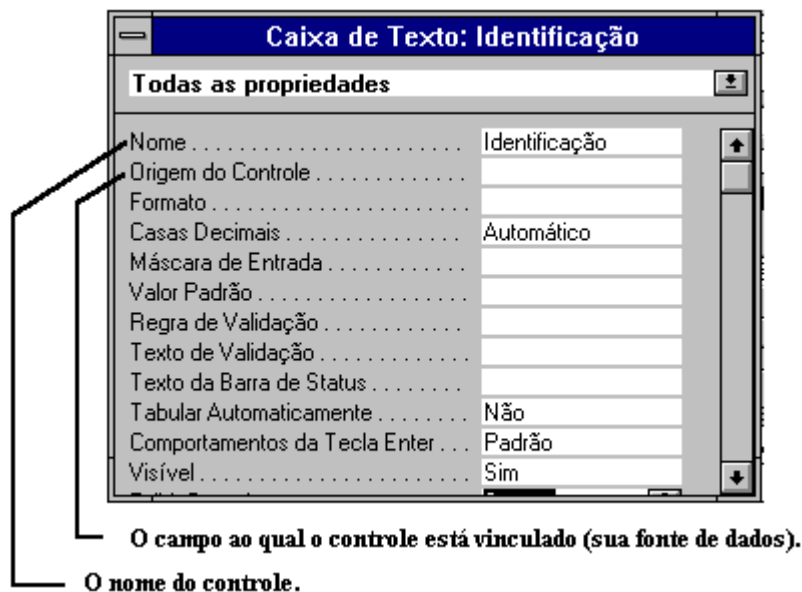
Para exibir a folha de propriedade de um controle

Clique duas vezes no controle (esse método funciona com todos os tipos de controle exceto sub-formulários, sub-relatórios e objetos OLE não vinculados).

-Ou-

Selecione o controle e escolha **Propriedades** no menu **Exibir** ou clique no botão “Propriedades” na barra de ferramentas.

Se o controle tem uma etiqueta anexada, selecione o controle e não sua etiqueta (para exibir a folha de propriedades de uma etiqueta, selecione a etiqueta).



Aqui teremos algumas propriedades, comuns para vários Objetos, essas propriedades podem ser definidas nos controles tanto para formulário quanto para relatório, visto que cada controle tem em média 20 a 30 propriedades teremos aqui somente algumas que são comuns a vários controles, como por exemplo.

MS-Access 2.0 Avançado

| Propriedade | Descrição |
|------------------------------|--|
| Origem do Controle | Nome de Campo ou Expressão para usar como fonte de dados para Controle |
| Tabular Automaticamente | Pode gerar ou não uma tabulação quando o último caracter permitido em uma máscara de entrada de uma caixa de texto é inserido. |
| Comportamento da Tecla Enter | Padrão troca de campo e Nova Linha no campo, cria uma linha em branco no mesmo campo para inserção. |
| Visível | Torna o objeto visível |
| Ativado | Habilitar controle no modo formulário |
| Parada de Tabulação | Permite tabulações no controle |
| Índice de Tabulação | Definir ordem de tabulação |
| Pode Ampliar | Controle ou seção expande automaticamente para caber no texto., |
| Pode Reduzir | Determina se o tamanho de uma sessão ou controle será Reduzido verticalmente para acomodar a quantidade de dados. |
| Ao Entrar | Macro ou função executada quando o controle recebe o primeiro foco |
| Ao Sair | Macro ou função executada quando o controle perde o foco no mesmo formulário. |

Para obter mais informações consulte **Definindo Propriedades** no Help do MS-Access.

Personalizando formulários

Você pode ser mais produtivo com formulários personalizados que se adaptem às suas necessidades particulares. Por exemplo, você pode personalizar que apresente listas de opções para sua escolha, adicionar um botão que torne fácil a abertura de um formulário relacionado e usar imagens em vez de texto em botões.

Alterando ordem de tabulação

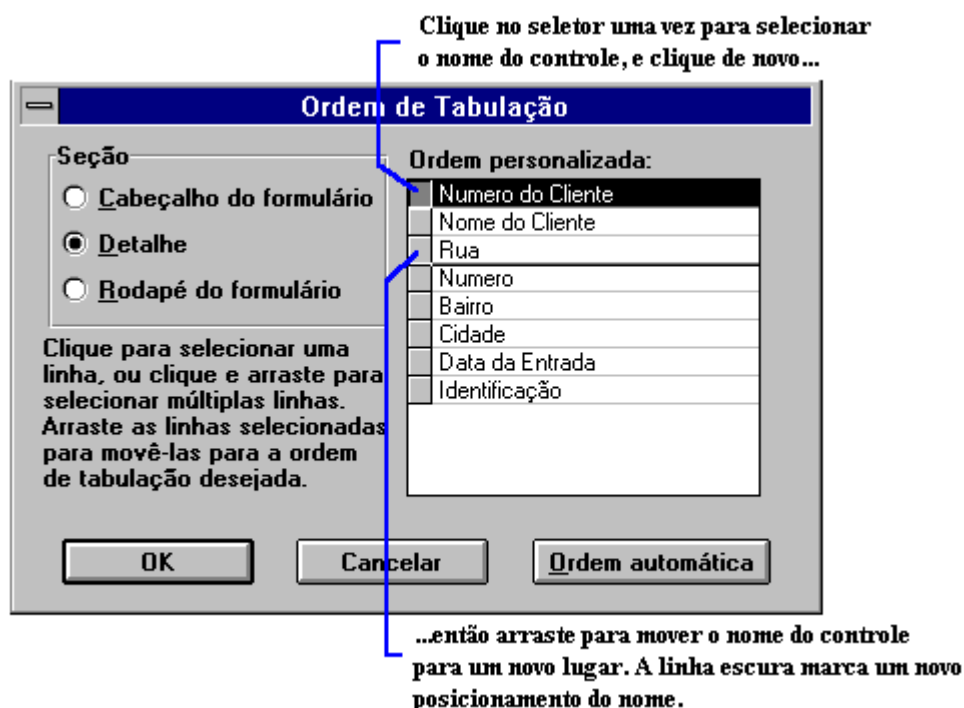
Quando adiciona controles a um formulário, o MS-Access atribui automaticamente uma ordem de tabulação aos controles. Quando a tecla Tab é pressionada no modo formulário o foco se move de controle para controle de acordo com a ordem de tabulação. O MS-Access coloca automaticamente no final da ordem de tabulação no final de cada novo controle adicionado. Em alguns casos essa pode não ser a ordem de tabulação desejada. Por exemplo, você pode reorganizar alguns controles em um formulário depois de colocá-los. Nesse caso, você provavelmente desejará alterar a ordem de tabulação para adequá-la à nova disposição. Para fazer isso, use o comando **Ordem de Tabulação** no modo Estrutura.

O comando **Ordem de Tabulação** exibe apenas os nomes dos controles que você pode selecionar ou para os quais pode-se tabular no modo formulário. Os nomes dos controles que não podem ser selecionados, como etiquetas ou linhas, não são incluídos. Os nomes são exibidos na ordem tabulada através dos controles. Você pode fazer com que o MS-Access crie automaticamente uma ordem de tabulação que percorra os controles da esquerda para a direita e de cima para baixo ou mesmo criar um ordem de tabulação personalizada.

Para criar uma ordem de tabulação da Esquerda para a Direita e de Cima para Baixo

1. Com o formulário aberto no modo estrutura, escolha **Ordem de Tabulação** no menu **Editar**.

2. Escolha o botão **Ordem Automática**.



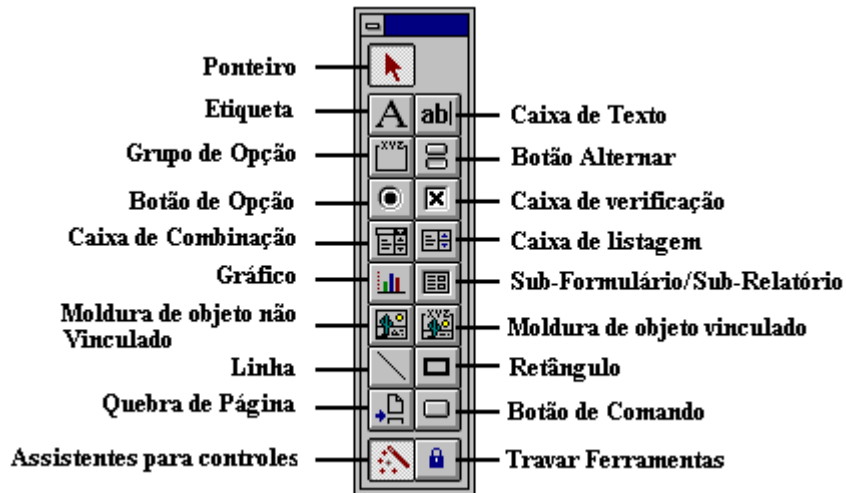
Criando outros controles utilizando a caixa de ferramentas

Se você deseja criar controles não vinculados ou calculados, use a barra de ferramentas. Também pode usar a caixa de ferramentas para criar controles vinculados que não sejam caixas de textos. A caixa de ferramentas é uma barra de ferramentas especializada: você pode escolher **Barras de Ferramentas** no menu **Exibir**, para ativar ou desativar a caixa de ferramentas.

O MS-Access exibe automaticamente a caixa de ferramentas quando você abre um formulário no modo estrutura, se as barras de ferramentas estiverem ativadas.

A caixa de ferramentas tem ferramentas para todos os tipos de controles que você pode criar na estrutura de um formulário.

Move-se a caixa de ferramentas selecionando e arrastando a mesma, e pode fechá-la clicando na caixa controle no canto superior esquerdo.



Para exibir ou fechar a caixa de ferramentas

No menu exibir, escolha caixa de ferramentas (ou clique no botão “caixa de ferramentas” na barra de ferramentas).

Para determinar a finalidade de uma ferramenta

Posicione o ponteiro do mouse sobre a ferramenta

O MS-Access exibe o nome da ferramenta.

Fornecendo uma lista de opções com uma caixa de listagem ou uma caixa de combinação

Uma caixa de listagem é uma lista simples de valores para você selecionar. Uma caixa de combinação é como uma caixa de texto e uma caixa de listagem combinadas em um controle, você tanto pode digitar um valor na caixa de texto como selecionar um valor na lista.

Vantagens da caixa de listagem

A lista aparece todas as vezes e o valor do controle fica limitado a o conjunto de alternativas da lista. Para mover rapidamente para o primeiro iniciado com uma determinada letra você pode digitar essa letra. Quando se está usando um formulário para inserir ou editar, não se pode adicionar um valor que não esteja na lista.

Vantagens de uma caixa de combinação

MS-Access 2.0 Avançado

Uma vez que a lista não é exibida até que se abra, o controle ocupa menos espaço no formulário. Você pode digitar os primeiros caracteres de um valor na caixa de combinação para rapidamente encontrar esse valor. Além disso, pode-se optar por permitir a entrada de valores novos na caixa de combinação ou impedir a entrada de valores novos.

Criando um Formulário/sub-formulário sem a ajuda do Assistente

Quando está criando um formulário e um sub-formulário sem a ajuda do assistente de Formulário/Sub-formulário, inicialmente você cria o formulário e o sub-formulário separadamente. Em seguida, insere o sub-formulário no formulário.

Este método é útil se você já criou um formulário e deseja inseri-lo em diversos outros formulários. Pode, também, preferir criar o seu próprio sub-formulário quando quiser criar um sub-formulário que seja diferente do formulário que o Assistente projetou.

Criando um formulário e um sub-formulário

Um sub-formulário é um formulário separado, que você cria e salva da mesma forma que qualquer outro formulário. Então, adiciona-o ao formulário de modo a se assegurar o seu vínculo ao formulário. Na maioria dos casos, o MS-Access vincula o formulário automaticamente quando você o adiciona. Existe, porém, situações onde você tem que definir as propriedades para vincular o formulário ao sub-formulário.

Preparando o formulário

1. Crie o formulário. Deixe espaço no formulário para o sub-formulário.
2. Salve e feche o formulário.

Preparando o sub-formulário

1. Crie um formulário e defina as propriedades de formulário *VisualizaçãoAutorizada* e *ModoPadrão*

Apenas folha de dados: Coloque os campos no formulário na ordem que você deseja que eles apareçam na folha de dados. Você não precisa organizar os campos, já que eles sempre aparecem como uma folha de dados.

Defina as propriedades *VisualizaçãoAutorizada* e *ModoPadrão* como folha de dados.

Apenas formulário: Coloque os controles como você deseja que apareçam no modo formulário. Defina a propriedade VisualizaçãoAutorizada como Formulário. Defina a propriedade ModoPadrão como formulário simples ou formulários contínuos.

Formulários e folha de dados: Coloque os controles como você deseja que eles apareçam no modo formulário. Defina a propriedade VisualizaçãoAutorizada para ambos. Defina a propriedade Modo Padrão para formulários Simples, formulários contínuos ou folha de dados. Dependendo de como você deseja que o sub-formulário seja visualizado quando você abre o formulário.

2. Salve e feche o sub-formulário.

Para adicionar um sub-formulário a um formulário

1. Abra o formulário no modo estrutura.
2. Alterne para a janela de Banco de Dados (Pressione F11).
3. Arraste o sub-formulário que quiser da janela Banco de Dados e coloque-o onde você o deseja no formulário.

O MS-Access adiciona o controle do sub-formulário a suas etiquetas anexadas ao formulário. O controle do sub-formulário está vinculado ao sub-formulário e é aproximadamente do mesmo tamanho do sub-formulário (o controle do sub-formulário inclui espaço para os seletores de registros, de forma que se torna maior).

4. Mova o controle do sub-formulário e, se necessário, ajuste o seu tamanho.
5. Alterne para o modo formulário para ver o formulário com seu sub-formulário.
6. Para fazer alterações no formulário, alterne para o modo Estrutura.

Para fazer alterações no sub-formulário, consulte o seguinte procedimento.

1. No modo estrutura, clique em qualquer lugar do formulário fora do controle do sub-formulário (para ter certeza de que o controle do sub-formulário não está selecionado).
2. Clique duas vezes em qualquer dentro do sub-formulário. (O MS-Access exibe o sub-formulário no modo estrutura).
3. Faça as alterações na estrutura do sub-formulário.
4. Salve e feche o sub-formulário no modo estrutura ou no modo folha de dados.

MS-Access 2.0 Avançado

5. Clique no botão “Modo do Formulário” na barra de ferramentas.

Vinculando um formulário e um sub-formulário

1. No formulário no modo Estrutura, abra a folha de propriedades para o controle do sub-formulário.

2. Defina a propriedade **VincularAPartirDe** como o nome do campo vinculado no sub-formulário.

Para entrar diversos campos vinculados, separe os nomes dos campos com ponto e vírgula.

3. Defina a propriedade **VincularA** como o nome do controle ou campo vinculado no formulário.

Para entrar diversos controles ou campos vinculados, separe os nomes de campo ou controle por ponto e vírgula.

4. Salve e feche o sub-formulário.

5. Clique no botão “Modo do Formulário” na barra de ferramentas.

Exercício V

1. Crie um Formulário para mostrar os pedidos feitos por um cliente.
2. Crie um Formulário para mostrar as informações do cliente
3. Crie um Formulário Principal/SubFormulário colocando no Principal as informações do cliente e no subFormulário os pedidos feitos por cada cliente

Calculando um total para um grupo de registros

Use a função **Soma** para calcular o total de um grupo de registros. As sintaxe para a função **Soma** é:

Soma(*expr*)

O argumento *expr* é o nome de um campo ou de uma expressão.

É possível usar nomes de campo no argumento *expr* da função **Soma**, mas não nomes de controles. Os nomes de campo podem vir de uma tabela ou de uma consulta. Pode-se até usar o nome de um campo calculado a partir de uma consulta. Entretanto, para saber o total dos valores em um controle calculado, é necessário repetir a expressão usada no controle calculado na função **Soma**.

Para acrescentar valores em
um controle calculado...

The screenshot shows a form with two sections. The top section is titled 'Detalhe' and contains a control labeled 'Valor com Desconto:' with the formula `= [Valor] - ([Valor] * 0,25)`. The bottom section is titled 'Rodapé do Formulário' and contains a control labeled 'Total:' with the formula `= Soma([Valor] - ([Valor] * 0,25))`. Two blue arrows point from the text above to the formula boxes, and another blue arrow points from the text below to the 'Total:' label.

...repita a expressão na função Soma.

Exercício VI

Crie um formulário tabular para mostrar todas os itens de um pedido e seus respectivos preços. No rodapé do formulário crie um campo somando todas os preços dos produtos.

Estruturando relatórios

Adicionando seções

Você pode aumentar a eficiência de um relatório adicionando *seções*, que são elementos próprios da estrutura do relatório. O MS-Access possui sete tipos de *seções*, que pode ser usado em qualquer combinações. Por exemplo, se você deseja imprimir informações introdutórias a um relatório, você adicionar um cabeçalho que imprima apenas na primeira página do relatório.

No modo estrutura, você vê a estrutura do relatório como *seções* e controles. O modo estrutura apresenta o modelo para o relatório, cada *seção* do relatório é representada uma vez, ainda que, no relatório impresso, algumas *seções* podem repetir algumas vezes. Por exemplo, quando você imprime um relatório, o cabeçalho do grupo e o rodapé do grupo se repetem para cada grupo, enquanto a *seção* detalhe se repete para cada registro.

Cabeçalho do Relatório

Aparece uma única vez no início do relatório. Você pode usá-lo para itens tais como logotipo. O cabeçalho do relatório é impresso antes do cabeçalho da página somente na primeira página do relatório.

Cabeçalho da Página

Aparece na parte superior de todas as páginas do relatório. É usado para exibir itens tais como legendas de colunas em relatório tabular.

Para adicionar cabeçalho e rodapé a um relatório ou página

No menu formatar, escolha cabeçalho/rodapé do relatório

Cabeçalho do Grupo

Aparece no início de um novo grupo de registros. É usado para exibir itens, tais como o nome do grupo.

Para adicionar cabeçalho e rodapé a um grupo

A ordem de classificação de um relatório é definida na caixa de diálogo **Classificar e Reagrupar**.

Para exibir a caixa de diálogo Classificar e Reagrupar

1. Abre um relatório no modo estrutura
2. No menu exibir escolha classificar e reagrupar ou clique no botão Classificar e Reagrupar na barra de ferramentas.

3. Na coluna campo/expressão seleciona o campo que você deseja ou digite uma expressão. (Tenha certeza de iniciar uma expressão com sinal de igual.)

Seção detalhe

Contém o corpo principal dos dados de um relatório. Os controles desta seção repetem-se para cada registro retirado do banco de dados.

Rodapé do grupo

Aparece no final de um grupo de registro. É usado para exibir itens, tais como os totais de grupo.

Rodapé da página

Aparece na parte inferior de todas as páginas do relatório. É usado para exibir itens tais como a numeração de páginas.

Rodapé do relatório

Aparece uma vez no final do relatório. É usado para exibir itens tais como totais de relatório. É a última seção na criação do relatório, mas aparece antes do rodapé da página na última página do rodapé impresso.

Exercício VII

Crie um relatório para mostrar uma saída como o padrão abaixo:

| Cabeçalho do Relatório | | | | | | | | | |
|---------------------------------|--|-----------------------------------|--|----------------|--|----------------------------|--|-----------|--|
| Empresa de Comércio de Produtos | | | | | | | | | |
| Listagem de Produtos | | | | | | | | | |
| Cabeçalho da Página | | | | | | | | | |
| Listagem de Produtos | | | | | | Pág: | | =[Página] | |
| Cabeçalho Nome da categoria | | | | | | | | | |
| categoria: | | Nome da categoria | | | | | | | |
| Cabeçalho Nome da companhia | | | | | | | | | |
| Fornecedor: | | Nome da companhia | | | | | | | |
| Detalhe | | | | | | | | | |
| Nome do produto | | | | Preço unitário | | Unidades em estoque | | | |
| Rodapé Nome da categoria | | | | | | | | | |
| | | Total de Produtos desta Categoria | | | | =Contar([Nome do Produto]) | | | |
| Rodapé da Página | | | | | | | | | |
| | | Produtos nesta página: | | | | =Contar([Nome do Produto]) | | | |
| Rodapé do Relatório | | | | | | | | | |
| | | Produtos listados: | | | | =Contar([Nome do Produto]) | | | |

Usando expressões em relatórios

Você usa expressões para obter informações que não consegue obter diretamente das tabelas de um banco de dados. Por exemplo, uma expressão pode ser utilizada para saber quantas unidades de um produto você vendeu em um mês ou o número médio de unidades que vende todo mês.

Um controle que tenha expressão igual a de sua origem de controle é um controle calculado. Se o controle é uma caixa de texto, você pode inserir a expressão diretamente na caixa de texto. De qualquer modo, pode sempre digitar a expressão diretamente na folha de propriedades ou pode definir a propriedade origem do controle com o construtor de expressões.

Macros

Usando macros, você fazer seus formulários, relatórios e outros objetos de banco de dados trabalharem juntos mais eficientemente. As macros podem automatizar tarefas rotineiras ou repetitivas tais como imprimir relatórios semanais. As macros também oferecem capacidade de personalização, tais como validação de dados complexas para formulários. Você pode até usar macros para criar suas próprias barras de menu personalizadas para seus formulários.

O que é uma macro?

Uma macro executa automaticamente uma tarefa ou um série de tarefas. Cada tarefa que deseja que o MS-Access execute é chamada uma ação. O MS-Access proporciona uma lista de ações dentre as quais seleciona-se para criar uma macro. Quando você executa a macro, o MS-Access executa as ações na sequência em que elas estão listadas, usando os objetos ou dados que tiver especificado. Por exemplo, você pode criar uma macro para abrir automaticamente uma tabela e um formulário usados juntos freqüentemente. Esta macro usa duas ações para dizer ao MS-Access para abrir a tabela e, então, abrir o formulário.

Quando se adiciona ou altera dados no formulário, pode-se usar uma outra macro para verificar os dados e exibir uma mensagem ou para abrir um outro formulário.

Porque usar macros?

Qualquer tarefa repetitiva ou rotineira efetuada no MS-Access é uma boa candidata a um macro. A automatização de tarefas rotineiras traz eficiência e precisão para seu banco de dados, porque uma macro executa a tarefa sempre do mesmo modo. Usando macros, é possível:

Fazer seus formulários e relatórios trabalhem juntos

Muitas vezes, será interessante usar dois formulário ao mesmo tempo para procurar dados relacionados. *Por exemplo*, você poderá usar um formulário para visualizar informações sobre seus Fornecedores e um outro para visualizar informações sobre seus produtos. Usando macros, você poderá adicionar um botão de comando ao formulário Fornecedores de modo a abrir automaticamente o formulário Produto e, assim, apresentar apenas os produtos daquele fornecedor. Poderá, também, adicionar um botão a um formulário para imprimir um relatório como, por exemplo, para imprimir uma fatura e uma etiqueta de endereçamento de um formulário usado para registrar novos pedidos.

Encontrar e filtrar registros automaticamente

As macros podem acelerar o processo para encontrar os registros que deseja ver. *Por exemplo*, você pode anexar macros a botões em um formulário de empregados para poder automaticamente filtrar registros em subconjuntos; por exemplo, todos os empregados localizados em Toronto ou todos os representantes de vendas que trabalham para você.

Definir valor em controles

Usando macros, você pode definir o valor de um controle em um formulário como o resultado de um cálculo ou como um valor de uma outra tabela. *Por exemplo*, quando você adiciona um produto a um formulário de pedidos, você pode usar uma macro para procurar automaticamente o preço unitário atual do produto e inserir o valor em um controle do formulário.

Garantir a precisão dos dados

As macros são ideais para validação e tratamento de dados em formulário. *Por exemplo*, você pode usar uma macro para responder a diferentes tipos de dados em um campo com diferentes mensagens. Suponha que você tenha clientes localizados em diversos países. Você pode usar uma macro para certificar-se de que o código postal inserido no registro de um novo cliente seja apropriado para o país onde vive o cliente.

Definir propriedade de formulários, relatórios e controles.

Você pode usar macros para definir a maioria das propriedades de formulários, relatórios e controles. *Por exemplo*, você pode usar uma macro para ocultar um formulário quando você não precisa vê-lo na tela, mas ainda precisa usar os dados do formulário.

Automatizar transferências de dados

Você pode usar uma macro para importar e exportar dados automaticamente entre diferentes formatos de arquivo. Se você exporta dados de vendas no final de cada semana para uma planilha MS-Excel. *Por exemplo*, você pode automatizar a tarefa em uma macro e fazer isso com o clique de um botão.

Criar o seu próprio ambiente de trabalho personalizado

Você pode usar uma macro para abrir automaticamente um grupo de consultas, formulários e relatórios quando você abre um banco de dados. Usando macros, você pode combinar partes individuais de um banco de dados em um aplicativo de banco de dados interativo; um programa que execute uma tarefa ou um conjunto de tarefas específicas tais como encontrar informações de vendas ou de faturas.

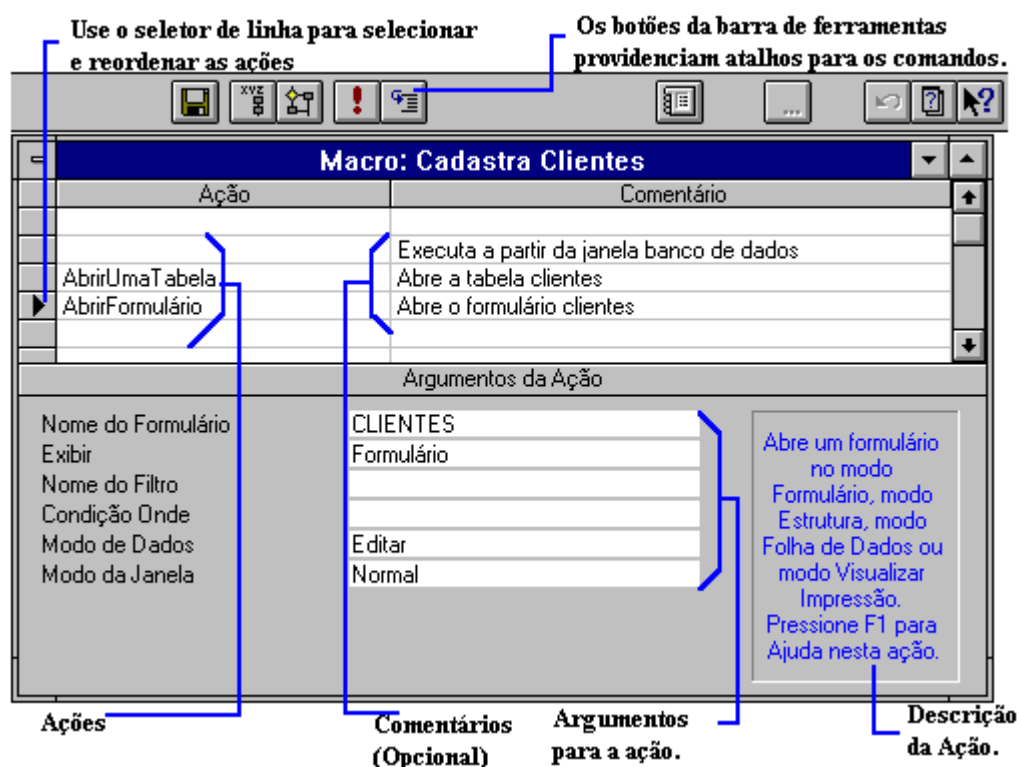
Criando uma Macro

Para criar uma macro

1. Na janela banco de dados, clique no botão “macro” (ou escolha macros no menu exibir)
2. Escolha o botão novo na janela banco de dados (ou escolha novo no menu arquivo e então escolha Macros)

A Janela Macro

Você usa a parte superior da janela macro para adicionar ações. Você pode incluir um comentário para explicar as razões por trás de cada ação (O MS-Access ignora comentários quando executa uma macro). Você usa a parte inferior da janela macro para especificar *argumentos* para uma ação. Os argumentos dão ao MS-Access informações adicionais sobre como executar uma ação, tal como qual objeto ou dados utilizar.



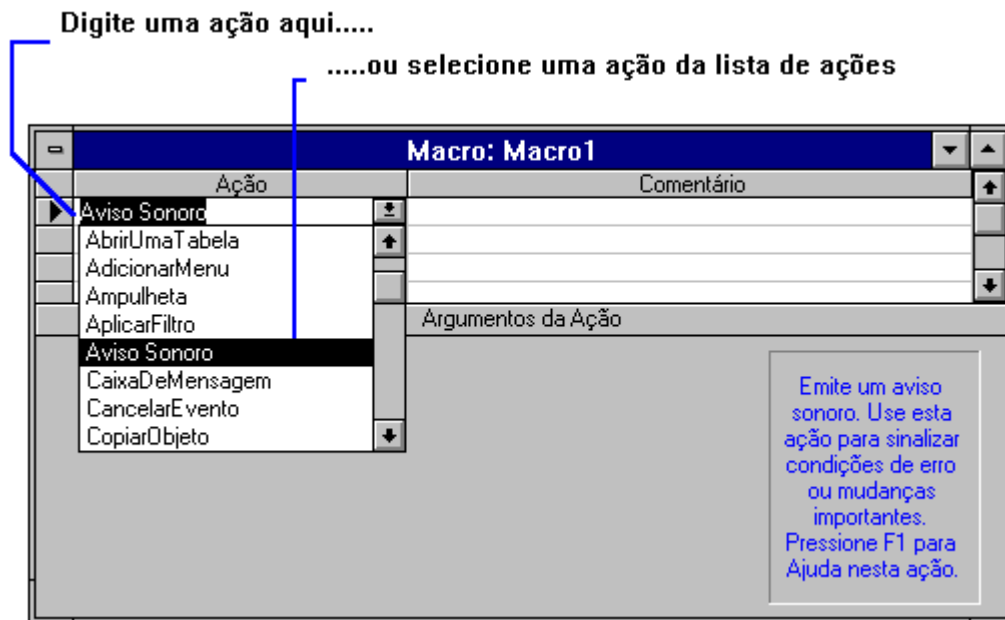
Adicionando ações a uma Macro

Existem duas maneiras de adicionar ações a uma macro. Você pode selecionar ações a partir da lista de ações da janela Macro ou, para ações comuns associadas a objetos de seu banco de dados, você pode arrastar o objeto da janela Banco de Dados para a célula ação na janela Macro. Esse método arrastar-e-soltar adiciona a ação e argumentos apropriados.

Para adicionar uma ação a uma macro

1. Na janela Macro, clique na primeira célula vazia da coluna Ação.
2. Selecione a ação desejada a partir da lista de ações (clique na seta para ver esta lista) ou digite a ação diretamente na célula.

Por exemplo: para que o computador execute um aviso sonoro, quando você executar uma macro, adicione a ação AvisoSonoro à macro.



3. Insira um comentário (opcional) sobre a ação.
4. Especifique os argumentos da ação, se eles forem necessários.

Para adicionar uma ação arrastando um objeto da janela banco de dados

1. Posicione a janela macro e a janela banco de dados de modo que ambas fiquem visíveis em sua tela (para redimensionar e posicionar rapidamente as janelas, escolha lado a lado no menu janela).

2. Na janela banco de dados, clique no botão para o ripo de objeto que deseja arrastar. Por exemplo, clique no botão “formulário” para arrastar um formulário.

3. Arraste o objeto a partir da janela banco de dados para uma linha de ação na janela macro.

- Se você arrastar uma tabela, consulta, formulário, relatório ou módulo para a janela Macro, o MS-Access adiciona uma ação que abre a tabela, consulta, formulário ou relatório.
- Se você arrastar uma macro para a janela Macro, o MS-Access adiciona uma ação que executa a macro.

Definindo argumentos

A maioria das ações tem argumentos que dão ao MS-Access informações adicionais sobre como executar a ação, tais como qual objeto ou dados utilizar. Por exemplo, a ação AbrirUmaTabela tem um argumento Nome da Tabela que informa ao MS-Access qual tabela do banco de dados abrir. Depois de adicionar uma ação, você define os argumentos para a ação na parte de baixo da janela Macro. Para definir um argumento, você pode digitar um valor na caixa de argumentos ou em muitos casos você pode selecionar o argumento apropriado de uma lista.

Muitas ações têm um argumentos que pede o nome de um objeto de banco de dados. Você pode usar uma técnica similar de “arrastar e largar” para definir estes e outros argumentos: você arrasta o objeto para a parte inferior da janela Macro e o larga no argumento que deseja definir. Por exemplo, suponha que você adicione a ação AbrirUmaTabela a uma macro inserindo a ação na linha de ação. Para definir o argumento Nome da Tabela da ação, você tanto pode selecionar a tabela desejada da lista do argumento Nome da Tabela, como você já pode arrastar a tabela da janela Banco de Dados para o argumento Nome da Tabela.

Salvando uma Macro

você deve salvar uma macro antes de executá-la. Quando você salva uma macro, ela se torna um objeto de banco de dados que você pode abrir ou executar a partir da janela Banco de Dados.

Criando um grupo de macro

Você às vezes pode achar conveniente agrupar algumas macros relacionadas em uma janela Macro. Para agrupar macros, você cria um grupo de macros. Um grupo de macro é um conjunto de uma ou mais macros que aparecem na mesma janela Macro. As próprias macros individuais são executadas independentes umas das outras.

Por exemplo, suponha que você tenha seis botões de comando em um formulário, cada um abrindo um formulário diferente. Em vez de ter seis macros separadas em seu banco de dados, uma para cada botão, você pode criar um grupo de macro que contenha uma macro para cada botão. Desta forma, você tem todas as macros relacionadas ao formulário juntas em um conveniente documento. Em vez de abarrotar a lista de macros da janela banco de dados com mais de seis macros, você adiciona somente uma.

Para criar um grupo de macro

1. Crie uma macro (na janela Banco de Dados, clique no botão “Macro” e, então, escolha o botão “Novo”).
2. Na janela Macro, escolha **Nome da Macro** no menu **Exibir** (ou clique no botão “Nome da Macro” na barra de ferramentas).
O MS-Access exibe a coluna Nome da Macro na parte superior da janela Macro.
3. Digite um nome da macro para cada conjunto de ações que compõem uma macro.

Todas as macros juntas formam o grupo de macros botões do Painel de Controle do Formulário.

Cada item na coluna Nome da Macro é uma macro individual.

Uma macro pode conter uma ação...

| Macro: Macro1 | | |
|---------------------|------------------|---|
| Nome da Macro | Ação | Comentário |
| | | Anexada aos botões do formulário P. de controle do formulário |
| Categorias | AbrirFormulário | Abre o formulário Categorias |
| Clientes | AbrirFormulário | Abre o formulário Clientes |
| Empregados | AbrirFormulário | Abre o formulário Empregados |
| Produtos | AbrirFormulário | Abre o formulário Produtos |
| | MoverDimensionar | Move o formulário Produtos |
| Representantes | AbrirFormulário | Abre o formulário Representantes de Vendas |
| Fornecedores | AbrirFormulário | Abre o formulário Fornecedores |
| Voltar ao Principal | AbrirFormulário | Volta para o formulário Painel de controle Principal |
| Argumentos da Ação | | |

...ou mais de uma ação.

Usando condições em macros

Em alguns casos, você pode querer executar uma ação ou uma série de ações em uma macro somente se uma determinada condição for verdadeira. Por exemplo, se você está usando uma macro para validar dados em um formulário, você pode querer exibir uma mensagem em resposta a um conjunto de valores inseridos em um registro e uma outra mensagem em resposta a um conjunto diferente de valores. Em casos como esses, você pode usar condições para controlar o fluxo da macro.

Uma condição é uma expressão lógica. A macro segue caminhos diferentes se a condição for verdadeira ou falsa. A tabela seguinte mostra exemplos de condições de macros.

| Condição | Resultado |
|--|--|
| Formulários![Estudantes]![Idade]>21 | Se o valor no controle Idade do Formulário Estudantes for maior que 21, então... |
| [Cidade]="Paris" | Se o valor no controle Cidade do formulário do qual a macro foi executada for Paris, então... |
| Formulários![Clientes]![País] = "USA" E Formulários![Clientes]![Código do Cliente] <31 | Se o valor no controle País do formulário Clientes for USA e o valor do controle Código do Cliente do formulário Clientes for menor que 31, então... |

Você insere condições na coluna Condição na janela Macro. Se uma condição for verdadeira, o MS-Access executa a ação naquela linha. Você pode fazer com que o MS-Access execute uma série de ações se a condição for verdadeira digitando elipses (...) na coluna Condições das ações que seguem imediatamente a condição.

Para adicionar uma condição a uma macro

1. Na janela Macro, escolha **Condição** no menu **Exibir** (ou clique no botão "Condição" na barra de ferramentas).

O MS-Access exibe a coluna Condição na janela macro.

2. Na coluna Condição, insira uma expressão condicional na linha onde você deseja definir a condição.

3. Na coluna Ação, insira a ação que você deseja que o MS-Access execute quando a condição for verdadeira.

Lista de ações de Macro

| Categoria | Descrição | Ação |
|-----------------------------------|---|---|
| Dados em formulários e relatórios | Restringir dados Mover através dos dados | AplicarFiltro EncontrarPróxima, EncontrarRegistro, IrParaControle, IrParaPágina, IrParaRegistro |
| Execução | Executar um comando Executar uma Macro, procedimento ou consulta Executar um outro aplicativo Parar execução | ExecutarItemDoMenu AbrirConsulta, ExecutarCódigo, ExecutarMacro, ExecutarSQL ExecutarAplicativo CancelarEvento, Sair, Parar Macro, PararTodasMacros |
| Importar/Exportar | Transferir dados entre o MS-Access e outros formatos de dados Enviar objetos do MS-Access para outro aplicativos | TransferirBancoDeDados, TransferirPlanilha, TransferirTexto SairPara, EnviarObjeto |

| Categoria | Descrição | Ação |
|------------------------|---|---|
| Manipulação de objetos | Definir o valor de um campo, controle ou propriedade | DefinirValor |
| | Atualizar dados ou a tela | RedesenharObjeto, RepetirAconsulta, ExibirTodosRegistros |
| | Selecionar um banco de dados | SelecionarObjeto |
| | Copiar ou Renomear um objeto | CopiarObjeto, Renomear |
| | Excluir um objeto | ExcluirObjeto |
| | Abrir ou fechar um objeto de banco de dados | Fechar, AbrirFormulário, AbrirMódulo, AbrirConsulta, AbrirRelatório, AbrirUmaTabela |
| | Imprimir um objeto de banco de dados | AbrirFormulário, AbrirConsulta, AbrirRelatório, Imprimir |
| Miscelânea | Mover ou redimensionar uma janela | Maximizar, Minimizar, MoverDimensionar, Restaurar |
| | Exibir informações na tela | Eco, Ampulheta, CaixaDeMensagem, DefinirAvisos |
| | Exibir ou ocultar a barra de ferramentas | ExibirBarraDeFerramentas |
| | Gerar pressionamento de teclas | EnviaSequênciaDeCaracteres |
| | Soar um aviso | AvisoSonoro |
| | Criar uma barra de menu personalizado para um formulário ou relatório | AdicionarMenu |

Respondendo a eventos em formulários

Quando você usa um formulário, o MS-Access reconhece certos acontecimentos no formulário como *eventos*. Por exemplo, mover de um registro para outro em formulários é um evento; bem como clicar duas vezes em um controle. Você pode usar uma macro para responder a um evento.

Fazendo Formulários trabalharem em conjunto

Se você tem um formulário com um sub-formulário em seu banco de dados, então você já viu uma maneira do MS-Access tornar possível diferentes formulários trabalharem juntos. Um sub-formulário é na realidade um formulário separado inserido em um formulário principal. Na maioria dos casos, você usa sub-formulários para mostrar registros relacionados a um registro do formulário principal. Por exemplo, um sub-formulário poderia conter uma lista de todas as partes integrantes de um produto manufaturado e o formulário principal poderia conter informações sobre o próprio produto. Quando você movesse para um novo registro no formulário principal, o sub-formulário passaria a mostrar os registros relacionados ao do formulário principal.

Usando macros, você pode mostrar registros relacionados de outra forma. Você pode mostrar registros relacionados em dois formulários ao mesmo tempo, cada um em sua própria janela. Isso dá a você a conveniência adicional de poder redimensionar, mover e fechar qualquer dos formulários independentes um do outro.

Exibir dados relacionados não é a única maneira de usar macros para fazer seus formulário trabalharem juntos. Você pode responder a um evento ocorrido em um formulário fazendo com que algo ocorra em outro formulário. Por exemplo, pode-se abrir um formulário, redimensioná-lo ou movê-lo, exibir ou ocultar controles e atualizar dados, entre outras coisas. É possível também abrir uma caixa de diálogo personalizada para coletar as informações necessárias para completar uma tarefa. Por exemplo, você poderia usar uma caixa de diálogo para perguntar qual dos vários relatórios diários imprimir.

Observação: Você pode desejar usar a ação RepetirAConsulta em uma macro para repetir a consulta de origem de um formulário, sub-formulário ou controle. Isso garante que os registros mais recentes sejam exibidos.

Atribuindo macros de validação a eventos de formulário e de controle

Para usar uma macro para validação de dados, você atribui a macro à propriedade de evento apropriada do formulário ou controle. As propriedades mais comumente usadas em validação de dados são *AntesDeAtualizar* e *AoExcluir*.

AntesDeAtualizar: Se Você especifica uma macro como definição da propriedade de *formulário* *AntesDeAtualizar*, a macro é executada quando o controle em um registro que te tenha sido alterado e vai para um controle em um registro diferente, mas antes do MS-Access salvar as alterações na tela. Isso costuma ser usado quando a validação envolve valores em mais de um campo do formulário.

Se você especifica uma macro como definição da propriedade de controle *AntesDeAtualizar*, a macro é executada quando o foco deixa um controle cujo valor tenha sido alterado, mas antes do MS-Access aceitar a alteração. Um exemplo é quando a validação em um controle requer mais de uma condição.

Os eventos *AntesDeAtualizar* tanto de controles como de formulários também ocorrem se você escolher o comando **Salvar Registro** do menu **Arquivo** quando o foco estiver em um registro que tenha sido alterado.

AoExcluir : Se você especifica uma macro como definição da propriedade de formulário *AoExcluir*, a macro é executada antes do MS-Access excluir um registro de uma tabela. Isso costuma ser usado tanto em verificação de integridade referencial, como para se certificar de que um cliente não tenha pedidos abertos, antes de excluir o registro do cliente. Embora o MS-Access tenha uma opção **Reforçar Integridade Referencial** incorporada, você pode não querer acionar essa opção em todas as ocasiões. Em tais casos, você pode usar uma macro especificada como definição da propriedade *AoExcluir* e obter resultados similares

É possível definir regras de validação usando macros ou definir a propriedade *RegraDeValidação* para controles e as propriedades *RegraDeValidação* para campos e registros. Estas são avaliadas na seguinte ordem quando você altera os dados em um controle de um formulário e, então, move para outro registros;

- A macro **AntesDeAtualizar do controle**
- A macro **AntesDeAtualizar do formulário**
- A expressão na propriedade **RegraDeValidação do controle**
- A expressão na propriedade **RegraDeValidação do campo de origem do controle vinculado**
- A expressão na propriedade **RegraDeValidação da tabela de origem**

Uma vez que você pode usar a ação **CancelarEvento** na macro de validação, pode-se usar a macro para uma validação preliminar; se a validação for bem sucedida, você pode, então, passar para a validação do controle, registros ou campo.

Exibindo mensagens de validação diferentes para condições diferentes

Vamos utilizar como exemplo uma empresa que importa produtos de fornecedores localizados em muitos países ao redor do mundo. Como países diferentes têm regras diferentes para seus códigos postais, você não pode simplesmente usar uma regra de validação para garantir que um valor inserido no controle **Código Postal** tenha as características corretas. Você pode, porém, escrever uma macro que verifique qual país está inserido no controle **País** e, então, verifique se o valor inserido no controle **Código Postal** tem as características certas para aquele país.

A macro tem duas condições. Para inserir as condições na janela **Macro**, clique no botão “**Condição**” na barra de ferramentas. Isso exibe a coluna **Condição** na janela **Macro**.

A primeira condição na macro verifica o código postal dos fornecedores localizados na França, Itália ou Espanha. Esses países têm códigos postais de 5 dígitos. Insira a seguinte expressão condicional:

[País] Em (“France”;”Itália”;”Spain”) E Compr([Código Posta])<>5

O operador **Em** compara o valor no controle **País** do formulário atual com a lista de países dentro dos parênteses. A função **Compr** conta o número de caracteres no controle **Código Postal**. Se um país estiver na lista e o número de caracteres do código postal não for igual a 5, o MS-Access executará a ação na mesma linha que a condição e quaisquer ações imediatamente a seguir que tenham reticências(...) na coluna **Condição**.

A seguir, adicione duas ações para executar quando a condição for atendida: a ação CaixaDeMensagem e a ação CancelarEvento. A tabela seguinte mostra os argumentos chave para a ação CaixaDeMensagem.

| Argumento | Definição | Descrição |
|-----------|--|---|
| Mensagem | O código Postal precisa ser de 5 caracteres. | A mensagem a exibir na caixa de mensagem |
| Tipo | Informação | O tipo de ícone a exibir com a mensagem |
| Título | Erro de Código Postal | O título que aparece no topo da caixa de mensagem |

A segunda condição na macro usa o mesmo tipo de expressão que a primeira, exceto que verifica o código postal do fornecedores localizados na Austrália e Singapura. Esses países têm códigos postais de 4 dígitos. Insira a seguinte expressão condicional:

[País]Em("Australia";Singapura")E Compr([Código Postal])<>4

Adicione as mesmas duas ações a serem executadas quando a condição for satisfeita mas, altere o argumento Mensagem da ação CaixaDeMensagem para que se leia: "O Código Postal precisa ser de 4 caracteres". Finalmente, adicione uma ação IrParaControle que mova o foco para controle Código Postal, facilitando a correção do erro.

Salve a macro e dê o nome Verificar Códigos Postais.

Macro: codigo postal

| Condição | Ação |
|---|-----------------|
| [País] Em ["France"; "Italia"; "Spain"] E Compr([Código Postal]) <> 5 | CaixaDeMensagem |
| ... | CancelarEvento |
| [País] Em ["Austrália"; "Singapura"] E Compr([Código Postal]) <> 4 | CaixaDeMensagem |
| ... | CancelarEvento |
| ([País]="Canadá") E ([Código Postal] Negado Como "[a-z][0-9]") | CaixaDeMensagem |
| ... | CancelarEvento |
| | IrParaControle |

Argumentos da Ação

| | |
|--------------|----------------------------------|
| Mensagem | O Código Postal precisa ser de 5 |
| Aviso Sonoro | Sim |
| Tipo | Nenhum |
| Título | Erro de Código Postal |

Entre uma expressão condicional nesta coluna.

Elipses.

Finalmente, defina a propriedade AntesDeAtulizar do formulário para Verificar Códigos Postais, o nome de sua macro. Você usa a propriedade do formulário em vez da propriedade do controle porque sua validação envolve dois controles diferentes no formulário.

Agora quando você usar o formulário para adicionar um novo fornecedor, o MS-Access executará sua macro de validação depois que você terminar de inserir o novo registro, mas antes de salvá-lo na tabela Fornecedores. Se alguma condição na macro for satisfeita, o MS-Access exibirá a mensagem apropriada e não salvará o registro na tabela Fornecedores. Em vez disso, o foco continuará no registro para que você possa corrigir os dados inválidos.

Criando nossos próprios menus

Para criar um menu personalizado temos que utilizar pelo menos duas macros, uma para definir o menu e outra para adicionar o menu.

Definição do Menu

1. Crie uma Macro nova;
2. Adicione a coluna de nomes à macro;
3. Na coluna nome digite o nome do menu, se desejar que uma letra fique sublinhada para ser tecla de acesso, antes da letra coloque o símbolo &;
4. Na coluna ação, coloque o que você deseja que a macro faça.

Veja o exemplo:

| Nome | Ação | Descrição |
|---------|----------------|-------------------------------|
| &Abrir | AbrirUmaTabela | Abre a tabela de Clientes |
| &Fechar | Fechar | Fecha o objeto ativo |
| - | | Adiciona um separador ao menu |
| &Sair | Sair | Sai do Access |

Quando terminar, salve a macro com o nome de MENU XXXX_YYYY, onde XXXX é a barra de menu a que este menu pertence, e YYYY é o nome do menu. Vejamos o exemplo:

Menu Principal_Arquivo

MS-Access 2.0 Avançado

Definição da Barra de Menu

1. Crie uma nova macro;
2. Adicione a coluna de nome à macro;
3. Na coluna Ação, escolha a ação AdicionarMenu
4. Nos argumentos da macro, coloque o nome do Menu (que irá aparecer na tela, vale também a regra do &), e o nome da macro que será ativada quando este menu for selecionado.

Salve a macro com o nome da Barra de menu que você deseja criar.

Atribuindo uma Barra de Menu a um formulário

Escolha o formulário que você deseja colocar a barra de menus, na Propriedade Barra de Menu, escolha o nome da macro que representa a sua barra de menus. Mude para o modo formulário e veja como ficou o menu.

Módulos

Para uma maior flexibilidade na manipulação dos seus dados, foi implementada no Access uma linguagem de programação chamada Access Basic, um dialeto do Visual Basic, que permite uma área para criação de rotinas vão complementar o seu banco de dados.

Estrutura do Módulo

Sub - SubFunção, é uma seqüência de comandos que executa alterações em formulários, tabelas ou qualquer outro objeto do MS-Access. Mas não é retornado nenhum valor para quem chamou esta **Sub**.

Function - Faz exatamente a mesma coisa que a Sub, mas tem obrigatoriamente que retornar um valor para que a chamou.

Declarando Variáveis

Existem dois tipos de declarações de variáveis, uma a nível de Módulo, que é visível em todas as Subs e Functions do Módulo e as que são visíveis apenas em numa função ou numa subfunção.

Declaração Global - É Utilizada no nível de módulo para declarar variáveis globais (variáveis disponíveis em todos os procedimentos em todos os módulos) e alocar espaço de armazenamento.

Declaração Dim - Utilizada no nível de módulo e no nível de procedimento para declarar as variáveis e alocar espaço de armazenamento. Quando utilizada a nível de módulo, as variáveis declaradas são visíveis a todos os procedimentos contidos no modulo. Se declaradas dentro de procedimentos, estas só serão visíveis no procedimento onde foram declaradas.

Tipos de Dados

A seguinte tabela mostra os tipos de dados fundamentais suportados pelo Access Basic, sufixo de declaração de tipo, tamanho dados e valores aceitáveis.

| Tipo de Dados | Sufixo | Tamanho dos dados | Valores aceitáveis |
|---|--------|----------------------|--|
| Integer | % | 2 | -32.768 a 32.767. |
| Long(inteiro longo) | & | 4 | -2.147.483.648 a 2.147.483.647. |
| Single(ponto flutuante de precisão simples) | ! | 4 | -3,402823E38 a -1,401298E-45 para valores negativos; 1,401298E-45 a 3,402823E38 para valores positivos; e 0. |
| Double (ponto flutuante de precisão dupla) | # | 8 | -1,79769313486232 E308 a -4,94065645841247 E-324 para valores negativos; 4,94065645841247 E-324 a 1,79769313486232 E308 para valores positivos; e 0. |
| Currency(inteiro dimensionado) | @ | 8 | -922.337.203.685.477,5808 a 922.337.203.685.477,5807. |
| String | \$ | 1 byte por caractere | 0 até aproximadamente 65.535 bytes. (Deve-se deixar uma margem para o armazenamento). |
| Variant | Nenhum | Conforme apropriado | Qualquer valor numérico até o intervalo de um tipo de dados Double ou qualquer texto de caractere. |

Vejamos um exemplo de como o modulo é estruturado:



Descrição dos itens utilizados na página anterior:

| | |
|---------------------------|--|
| Function | Declara uma função, um bloco de comandos que ao seu término retorna um valor, neste exemplo, é retornado uma sequência de caracteres |
| Maiúsculas | Nome da função, este nome pode ser definido pelo usuário |
| (cTexto As String) | cTexto é o nome de uma variável que é passada para esta função; As String identifica o tipo de dado que esta variável contém. |
| As String | Informa o tipo de dado que é retornado pela função |
| Dim | Declara uma ou mais variáveis |
| String | Vide Tipos de Dados |
| Ucase\$() | Função padrão do Access Basic, torna todos os caracteres passados a ela em caracteres Maiúsculos |

Para retornarmos um valor de qualquer função definida pelo usuário, devemos atribuir o valor de retorno a função, como vemos na penúltima linha deste módulo.

Usando dados contidos nos Formulários

A coleção **Forms** contém todos os objetos Formulários abertos em um banco de dados do Microsoft Access.

Use a coleção Forms no Access Basic ou uma expressão para se referir a um formulário em particular.

Por exemplo, você pode usar a coleção Forms para verificar ou definir o valor de um controle de formulário. O objeto Formulário contém todas as propriedades da folha de propriedades de um formulário. Você pode se referir a um formulário das seguintes maneiras:

Forms![MeuFormulário]
Forms("MeuFormulário")

NomeEmp=Forms![Empregados]![Sobrenome] & ", " & Forms![Empregados]![Primeiro nome]

No exemplo acima, a variável NomeEmp recebe o conteúdo do controle **Sobrenome** mais o conteúdo do controle **[Primeiro nome]**, ambos contidos no formulário **Empregados**.

Podemos utilizar a coleção forms para testar condições em uma macro, neste caso, quando você pressionar a tecla Enter terminando a condição, a palavra Forms, será substituída pela palavra Formulário, não implicando em nenhum tipo de erro.

| Condição | Ação | Descrição |
|------------------------------------|-----------------|---|
| Forms![Produtos]!Descontinuado=SIM | CaixaDeMensagem | Se não tiver mais produtos exibe uma mensagem de erro |

Objetos de Banco de Dados

O Access possui vários objetos e coleções de objetos para tratamento de banco de dados, vamos discuti-los um a um.

| Objeto | Descrição |
|------------------|---|
| DBEngine | O DBEngine representa o sistema de recuperação e armazenamento de dados do Access, ele tem controle sobre todos os outros objetos |
| Workspace | Um objeto Workspace define uma sessão para um usuário. Contém bancos de dados abertos e provém mecanismos para transações simultâneas e para um grupo de trabalho seguro. |
| Database | Banco de Dados em si, um arquivo MDB, com suas tabelas, consultas e etc... |
| TableDef | Contem todos os itens de uma tabela, campos e índices. |
| Field | Contem as definições para um campo |
| Index | Contem as definições para um índice, campo, único e etc... |
| QueryDef | Definições de uma consulta, qualquer tipo |
| RecordSet | Conjunto de registros de uma tabela |
| Relation | O objeto Relation representa um relacionamento entre duas tabelas |

Todos os itens em **negrito** representam uma coleção de objetos, a diferença é que a coleção é no plural e o objeto tem o nome no singular. Para sabermos o nome dos campos de uma tabela, devemos utilizar a coleção **fields**.

Processo para trabalhar com o banco de dados atual:

01 Function Exemplo

```

02 Dim db As Database      'Define uma variável do tipo Database
03 Dim i As Integer        'Define uma variável do tipo inteiro

05 Set db = DBEngine(0)(0) ' Atribui o Banco de dados atual a db

06 For i = 0 to db.TableDefs.Count - 1 ' conta até a última tabela
07     Debug.Print db.TableDefs(i).Name 'Imprime o nome de cada
tabela
08 Next

```

MS-Access 2.0 Avançado

09 **db.Close**

10 **End Function**

No exemplo da página anterior, uma variável do tipo Database é declarada (02), na linha 05 é atribuída a ela o banco de dados atual. Nas linhas seguintes é feito um laço contando todas as tabelas que existem no banco de dados atual (06), na linha 07 é utilizado o objeto Debug para imprimir na Janela Imediata, o nome da tabela atual apontada pela coleção TableDefs, TableDefs é um vetor contendo todas as tabelas do banco de dados, a linha 08 fecha o laço e a 09 fecha o banco de dados db.

Experimente criar um módulo novo e digite a função acima, quando terminar, escolha no menu executar a opção Compilar módulos carregados para verificar se não existe nenhum erro de digitação.

Se não houver erro, posicione o cursor na primeira linha de código após a declaração Function e pressione F9 para gerar um ponto de parada. Minimize a janela.

Na janela de Banco de dados selecione o objeto Macro, crie uma nova macro para executar a nossa função. Execute-a.

Quando o Access atingir o ponto de parada, a janela de código será exibida para que possamos executar a função passo-a-passo pressionando a tecla F8. No menu Exibir escolha a opção Janela imediata para podermos ver a saída da nossa função.

Validação de campos com módulos

Para cada campo num formulário, podemos fazer validações, utilizado o evento AntesDeAtualizar, podemos fazer validações para os campos.

Veja o exemplo dado para o evento AntesDeAtualizar para o campo Nome do Contato:

```
Sub Nome_do_contato_BeforeUpdate (Cancel As Integer)
  Const MB_STOP_BUTTON = 16
  If IsNull([Nome do Contato]) Or IsEmpty([Nome do Contato]) Then
    MsgBox "Este campo não pode ficar vazio", MB_STOP_BUTTON,
    "Validação de Campo"
    Cancel = True
  End If
End Sub
```

No exemplo acima é verificado se o campo [Nome do Contato] é nulo ou está em branco, se ele for um dos dois, a caixa de mensagem será exibida com o símbolo de parada crítica.

Se for atribuído o valor verdadeiro (true) para a variável Cancel, o evento AntesDeAtualizar será cancelado, o campo não será atualizado e o foco permanecerá no campo [Nome do Contato].