Structured Query Language

SQL

Data Manipulation Language

DML

Linguagem de Manipulação de Dados

Parte II

Prof. André Luiz Alves

1. Operação INNER JOIN

Combina registros de duas tabelas sempre que houver valores correspondentes em um campo comum.

1.1. Sintaxe

FROM tabela1 INNER JOIN tabela2 ON tabela1.campo1 opercomp tabela2.campo2

1.2. A operação INNER JOIN possui as partes a seguir:

Parte	Descrição
tabela1, tabela2	Os nomes das tabelas das quais os registros são combinados.
campo1, campo2	Os nomes dos campos que são associados. Se não forem numéricos, os campos deverão ser do mesmo tipo de dados e conter o mesmo tipo de dados, mas não precisarão ter o mesmo nome.
opercomp	Qualquer operador de comparação relacional: "=," "<," ">," "<=," ">=," ou "<>."

1.3. Comentários

Você pode usar uma operação INNER JOIN em qualquer cláusula FROM. Este é o tipo mais comum de associação. As associações internas combinam registros de duas tabelas sempre que houver valores correspondentes em um campo comum a ambas.

Você pode usar INNER JOIN com as tabelas Departamentos e Funcionários para selecionar todos os funcionários em cada departamento. Em contrapartida, para selecionar todos os departamentos (mesmo que alguns não tenham funcionários designados para eles) ou todos os funcionários (mesmo que alguns não sejam designados a um departamento), você pode usar uma operação LEFT JOIN ou RIGHT JOIN para criar uma associação externa.

Se você tentar associar campos contento dados de Memorando ou Objeto OLE, ocorrerá um erro.

Você pode associar quaisquer dois campos numéricos de tipos semelhantes. Por exemplo, você pode associar em campos AutoNumeração e Longo pois são tipos semelhantes. Entretanto, você não pode associar os tipos de campo Único e Duplo.

O exemplo a seguir mostra como você poderia associar as tabelas Categorias e Produtos no campo CódigoDaCategoria:

```
SELECT NomeDaCategoria, NomeDoProduto
        FROM Categorias
        INNER JOIN Produtos
                ON Categorias.CódigoDaCategoria = Produtos.CódigoDaCategoria;
```

No exemplo acima, CódigoDaCategoria é o campo associado, mas não é incluído na saída da consulta pois não está incluído na instrução SELECT. Para incluir o campo associado, inclua o nome do campo na instrução SELECT ¾ neste caso, Categorias.CódigoDaCategoria.

Você pode também vincular várias cláusulas ON em uma instrução JOIN, usando a sintaxe a seguir:

```
SELECT campos
        FROM tabela1 INNER JOIN tabela2
        ON tabela1.campo1 opercomp tabela2.campo1 AND
        ON tabela1.campo2 opercomp tabela2.campo2) OR
        ON tabela1.campo3 opercomp tabela2.campo3)];
```

Você pode também aninhar instruções JOIN usando a seguinte sintaxe:

```
SELECT campos
    FROM tabela
    INNER JOIN (tabela2 INNER JOIN [( ]tabela3
    [INNER JOIN [( ]tabelax [INNER JOIN ...)]
    ON tabela3.campo3 opercomp tabelax.campox)]
    ON tabela2.campo2 opercomp tabela3.campo3)
    ON tabela1.campo1 opercomp tabela2.campo2;
```

Uma instrução LEFT JOIN ou RIGHT JOIN pode ser aninhada dentro de uma INNER JOIN, mas uma INNER JOIN não pode ser aninhada dentro de uma LEFT JOIN ou de uma RIGHT JOIN.

2. Operações LEFT JOIN, RIGHT JOIN

Combina registros da tabela de origem quando usados em qualquer cláusula FROM.

2.1. Sintaxe

```
FROM tabela1 [ LEFT | RIGHT ] JOIN tabela2
ON tabela1.campo1 opercomp tabela2.campo2
```

2.2. As operações LEFT JOIN e RIGHT JOIN possuem as partes a seguir:

Parte	Descrição
tabela1, tabela2	Os nomes das tabelas das quais os registros são combinados.
campo1, campo2	Os nomes dos campos que são associados. Os campos devem ser do mesmo tipo de dados e conter o mesmo tipo de dados, mas não precisam ter o mesmo nome.
opercomp	Qualquer operador de comparação relacional: "=," "<," ">," "<=," ">=," ou "<>."

2.3. Comentários

Utilize uma operação LEFT JOIN para criar uma associação externa esquerda. As associações externas esquerdas incluem todos os registros da primeira (esquerda) de duas tabelas, mesmo que não haja valores correspondentes para os registros na segunda tabela (direita).

Utilize uma operação RIGHT JOIN para criar uma associação externa direita. As associações externas direitas incluem todos os registros da segunda (direita) de duas tabelas, mesmo que não haja valores correspondentes para registros na primeira (esquerda) tabela.

Por exemplo, você poderia usar LEFT JOIN com as tabelas Departamentos (esquerda) e Funcionários (direita) para selecionar todos os departamentos, inclusive os que não tenham funcionários designados para eles. Para selecionar todos os funcionários, inclusive os que não estão designados para um departamento, você usaria RIGHT JOIN.

O exemplo a seguir mostra como você poderia associar as tabelas Categorias e Produtos no campo CódigoDaCategoria. A consulta produz uma lista de todas as categorias, inclusive aquelas que não contêm produtos:

```
SELECT NomeDaCategoria, NomeDoProduto
    FROM Categorias
    LEFT JOIN Produtos
     ON Categorias.CódigoDaCategoria = Produtos.CódigoDaCategoria;
```

Neste exemplo, CódigoDaCategoria é o campo associado, mas não é incluído nos resultados da consulta pois não está incluído na instrução SELECT. Para incluir o campo associado, digite o nome do campo na instrução SELECT ¾ neste caso, Categorias.CódigoDaCategoria.

Para criar uma consulta que inclua somente registros nos quais os dados nos campos associados sejam os mesmos, utilize uma operação INNER JOIN.

Uma operação LEFT JOIN ou RIGHT JOIN pode ser aninhada dentro de uma operação INNER JOIN, mas uma INNER JOIN não pode ser aninhada dentro de uma LEFT JOIN ou de uma RIGHT JOIN. Consulte a matéria sobre como aninhar no tópico INNER JOIN para ver como aninhar associações dentro de outras associações.

Você pode vincular várias cláusulas ON. Consulte a matéria sobre vinculação de cláusulas no tópico INNER JOIN para ver como isso é feito.

Se você tentar associar campos contendo dados de Memorando ou Objeto OLE, ocorrerá um erro.

3. Operação UNION

Cria uma consulta união, que combina os resultados de duas ou mais consultas ou tabelas independentes.

3.1. Sintaxe

[TABLE] consulta1 UNION [ALL] [TABLE] consulta2 [UNION [ALL] [TABLE] consultan [...]]

3.2. A operação UNION possui as partes a seguir:

Parte	Descrição
consulta1-n	Uma instrução SELECT, o nome de uma consulta armazenada ou o nome de uma tabela armazenada precedida da palavra-chave TABLE.

Você pode mesclar os resultados de duas ou mais consultas, tabelas e instruções SELECT, em qualquer combinação, em uma única operação UNION. O exemplo a seguir mescla uma tabela existente denominada Novas Contas com uma instrução SELECT:

```
TABLE [Novas Contas] UNION ALL
SELECT *
        FROM Clientes
        WHERE QuantiaDoPedido > 1000;
```

Como padrão, nenhum registro duplicado é retornado quando você usa uma operação UNION; entretanto, você pode incluir o atributo ALL para assegurar que todos os registros sejam retornados. Isso faz com que a execução da consulta seja mais rápida.

Todas as consultas em uma operação UNION devem solicitar o mesmo número de campos; contudo, os campos não deverão ter o mesmo tamanho ou tipo de dados.

Use aliases somente na primeira instrução SELECT pois eles são ignorados em qualquer outra. Na cláusula ORDER BY, refira-se aos campos pelo que são chamados na primeira instrução SELECT.

Você pode usar uma cláusula GROUP BY ou HAVING em cada argumento consulta para agrupar os dados retornados.

Você pode usar uma cláusula ORDER BY no fim do último argumento consulta para exibir os dados retornados em uma ordem especificada.

4. Subconsultas SQL

Uma subconsulta é uma instrução SELECT aninhada dentro de uma instrução SELECT, SELECT...INTO, INSERT...INTO, DELETE ou UPDATE ou de uma outra subconsulta.

4.1. Sintaxe

Você pode usar três formas de sintaxe para criar uma subconsulta:

comparação [ANY | ALL | SOME] (instruçãosql)
expressão [NOT] IN (instruçãosql)
[NOT] EXISTS (instruçãosql)

4.2. Uma subconsulta possui as partes a seguir:

Parte	Descrição
comparação	Uma expressão e um operador de comparação que compara a expressão com os resultados da subconsulta.
expressão	Uma expressão para a qual o conjunto de resultados da subconsulta é procurado.
instruçãosql	Uma instrução SELECT, que segue o mesmo formato e regras de qualquer outra instrução SELECT. Deve estar entre parênteses.

4.3. Comentários

Você pode usar uma consulta em lugar de uma expressão na lista de campos de uma instrução SELECT ou ema cláusula WHERE ou HAVING. Em uma subconsulta, você usa uma instrução SELECT para proporcionar um conjunto de um ou mais valores específicos para avaliar na expressão de cláusula WHERE ou HAVING.

Use o atributo ANY ou SOME, que são sinônimos, para recuperar registros na consulta principal que satisfaçam a comparação com qualquer registro recuperado na subconsulta. O exemplo a seguir retorna todos os produtos cujo preço unitário é maior que o de qualquer produto vendido com um desconto de 25% ou maior:

```
SELECT * FROM Produtos
        WHERE PreçoUnitário > ANY
        (SELECT PreçoUnitário FROM DetalhesDoPedido
        WHERE Desconto >= .25);
```

Use o atributo ALL para recuperar somente os registros da consulta principal que satisfaçam a comparação com todos os registros recuperados na subconsulta. Se você tivesse alterado ANY para ALL no exemplo anterior, a consulta retornaria somente os produtos cujo preço unitário fosse maior que o de todos os produtos vendidos com um desconto de 25% ou maior. Isso é bem mais restritivo.

Use o atributo IN para recuperar somente os registros da consulta principal para os quais algum registro na subconsulta contenha um valor igual. O exemplo a seguir retorna todos os produtos com um desconto de 25% ou maior:

```
SELECT * FROM Produtos
WHERE CódigoDoProduto IN
(SELECT CódigoDoProduto FROM DetalhesDoPedido
WHERE Desconto >= .25);
```

Da mesma forma, você pode usar NOT IN para recuperar somente os registros na consulta principal para os quais nenhum registro na subconsulta contenha um valor igual.

Use o atributo EXISTS (com a palavra reservada NOT opcional) em comparações verdadeiro/falso para determinar se a subconsulta retorna algum registro.

Você pode também usar aliases de nome de tabela em uma consulta para se referir a tabelas relacionadas em uma cláusula FROM fora da subconsulta. O exemplo a seguir retorna os nomes dos funcionários cujos salários são iguais ou maiores do que o salário médio de todos os funcionários que têm o mesmo cargo. A tabela Funcionários recebe o alias "T1":

```
SELECT Sobrenome, Nome, Título, Salário
        FROM Funcionários AS T1
        WHERE Salário >= (SELECT Avg(Salário)
                FROM Funcionários
        WHERE T1.Título = Funcionários.Título) Order by Título;
```

No exemplo anterior, a palavra reservada AS é opcional.

Algumas subconsultas são permitidas em consultas de tabela de referência cruzada — especificamente, como atributos (os da cláusula WHERE). Subconsultas como saída (as da lista SELECT) não são permitidas em consultas de tabela de referência cruzada.

5. Instrução TRANSFORM

Cria uma consulta tabela de referência cruzada.

5.1. Sintaxe

```
TRANSFORM funçãoagrg
        instruçãoselect
                PIVOT campocentral [IN (valor1[, valor2[, ...]])]
```

5.2. A instrução TRANSFORM possui as partes a seguir:

Parte	Descrição
funçãoagrg	Uma função agregada SQL que opera sobre os dados selecionados.
instruçãoselect	Uma instrução SELECT.
campodinâmicol	O campo ou expressão que você quer usar para criar títulos de coluna no conjunto de resultados da consulta.
valor1, valor2	Valores fixos usados para criar títulos de colunas.

5.3. Comentários

Quando resume dados utilizando uma consulta tabela de referência cruzada, você seleciona valores de campos ou expressões especificadas como títulos de colunas para poder visualizar os dados em um formato mais compacto do que com uma consulta seleção.

A instrução TRANSFORM é opcional, mas quando for incluída será a primeira instrução em uma seqüência SQL. Ela antecede uma instrução SELECT que especifica os campos usados como títulos de linhas e uma cláusula GROUP BY que especifica o agrupamento de linhas. Opcionalmente, você pode incluir outras cláusulas, como WHERE, que especifiquem critérios adicionais de seleção ou de classificação. Você pode também usar subconsultas como atributos — especificamente, aqueles em uma cláusula WHERE — em uma consulta tabela de referência cruzada.

Os valores retornados em campodinâmico são usados como títulos de colunas no conjunto de resultados da consulta. Por exemplo, articular os números das vendas no mês da venda em uma consulta tabela de referência cruzada criaria 12 colunas. Você pode restringir campodinâmico para criar títulos a partir de valores fixos (valor1, valor2) relacionados na cláusula IN opcional. Você pode também incluir valores fixos para os quais não existam dados para criar colunas adicionais.

6. Declaração PARAMETERS

Declara o nome e o tipo de dados de cada parâmetro em uma consulta parâmetro.

6.1. Sintaxe

PARAMETERS nome tipodedados [, nome tipodedados [, ...]]

6.2. A declaração PARAMETERS possui as partes a seguir:

Parte	Descrição
nome	O nome do parâmetro. Atribuído à propriedade Name do objeto Parameter e usado para identificar esse parâmetro na coleção Parameters. Você pode usar nome como uma seqüência de caracteres que é exibida em uma caixa de diálogo enquanto o aplicativo executa a consulta. Use colchetes ([ ]) para envolver o texto que contém espaços ou pontuação. Por exemplo, [Preço baixo] e [Começar relatório com que mês?] são argumentos nome válidos.
tipodedados	Um dos tipos de dados SQL do Microsoft Jet primários ou seus sinônimos.

6.3. Comentários

Para consultas executadas regularmente, você pode utilizar uma declaração PARAMETERS para criar uma consulta parâmetro. Uma consulta parâmetro pode ajudar a automatizar o processo de alteração dos critérios da consulta. Em uma consulta parâmetro, o código precisará fornecer os parâmetros a cada vez que a consulta for executada.

A declaração PARAMETERS é opcional, mas quando incluída precede qualquer outra instrução, inclusive SELECT.

Se a declaração incluir mais de um parâmetro, separe-os com vírgulas. O exemplo a seguir inclui dois parâmetros:

```
PARAMETERS [Preço baixo] Currency, [Data inicial] DateTime;
```

Você pode usar nome, mas não tipodedados em uma cláusula WHERE ou HAVING. O exemplo a seguir espera que dois parâmetros sejam fornecidos e, então, aplica os critérios aos registros na tabela Pedidos:

```
PARAMETERS [Preço baixo] Currency, [Data inicial] DateTime;
SELECT NúmeroDoPedido, QuantiaDoPedido
        FROM Pedidos
        WHERE QuantiaDoPedido > [Preço baixo] AND DataDoPedido >= [Data inicial];
```

7. Operador Between...And

Determina se o valor de uma expressão se situa dentro de um intervalo especificado de valores. Você pode utilizar este operador em instruções SQL.

7.1. Sintaxe

expr [Not] Between valor1 And valor2

7.2. A sintaxe do operador Between...And possui as partes a seguir:

Parte	Descrição
expr	Expressão que identifica o campo que contém os dados a serem avaliados.
valor1, valor2	Expressões em relação as quais você deseja avaliar expr.

7.3. Comentários

Se o valor de expr estiver entre valor1 e valor2 (inclusive), o operador Between...And retornará True; caso contrário, retornará False. Você pode incluir o operador lógico Not para avaliar a condição oposta (isto é, se expr estiver situado fora do intervalo definido por valor1 e valor2).

Você poderia utilizar Between...And para determinar se o valor de um campo está situado em um intervalo numérico especificado. O exemplo a seguir determina se um pedido foi enviado a um local situado em um intervalo de códigos postais. Se o código postal estiver entre 98101 e 98199, a função IIf retornará "Local". Caso contrário, retornará "Nãolocal".

SELECT IIf(CódigoPostal Between 98101 And 98199, "Local", "Nãolocal") FROM Editores

Se expr, valor1 ou valor2 forem Null, Between...And retornará um valor Null.

Uma vez que os caracteres curinga, como *, são tratados como literais, você não pode utilizá-los com o operador Between...And. Por exemplo, você não pode utilizar 980* e 989* para localizar todos os códigos postais que começam com 980 e 989. Em vez disso, você tem duas alternativas: pode adicionar uma expressão para a consulta que pegue os três caracteres da esquerda do campo de texto e utilizar Between...And nesses caracteres, ou pode preencher os valores superior e inferior com caracteres extras — neste caso, 98000 a 98999, ou 98000 a 98999 – 9999 se estiver utilizando códigos postais estendidos. (Você deve omitir o – 0000 dos valores inferiores pois, caso contrário, 98000 será excluído se alguns códigos postais tiverem seções e outros não.)

8. Operador In

Determina se o valor de uma expressão é igual a algum dos vários valores em uma lista especificada.

8.1. Sintaxe

expr [Not] In(valor1, valor2, . . .)

8.2. A sintaxe do operador In possui as partes a seguir:

Parte	Descrição
expr	Expressão que identifica o campo que contém os dados a serem avaliados.
valor1, valor2	Expressão ou lista de expressões em relação a qual você deseja avaliar expr.

Se expr for encontrado na lista de valores, o operador In retornará True; caso contrário, retornará False. Você pode incluir o operador lógico Not para avaliar a condição oposta (isto é, se expr não está na lista de valores).

Por exemplo, você pode utilizar In para determinar os pedidos a serem enviados a um conjunto de regiões especificadas:

```
SELECT *
        FROM Pedidos
        WHERE RegiãoDeRemessa In ('Avon','Glos','Som')
```

9. Operador Like

Compara uma expressão de seqüência com um padrão em uma expressão SQL.

9.1. Sintaxe

expressão Like "padrão"

9.2. A sintaxe do operador Like possui as partes a seguir:

Parte	Descrição
expressão	Expressão SQL utilizada em uma cláusula WHERE.
padrão	Seqüência ou literal de seqüência de caracteres em relação a qual expressão é comparada.

9.3. Comentários

Você pode utilizar o operador Like para localizar valores em um campo que correspondam ao padrão especificado. Para padrão, você pode especificar o valor completo (por exemplo, Like "Smith") ou pode utilizar caracteres curinga para encontrar um intervalo de valores (por exemplo, Like "Sm*").

Em uma expressão, você pode utilizar o operador Like para comparar um valor de campo a uma expressão de seqüência. Por exemplo, se você digitar Like "C*" em uma consulta SQL, a consulta retornará todos os valores de campo que começam com a letra C. Em uma consulta parâmetro, você pode solicitar ao usuário que forneça um padrão pelo qual procurar.

O exemplo a seguir retorna dados que começam com a letra P, seguida por qualquer letra entre A e F e três dígitos:

Like "P[A-F]###"

A tabela a seguir mostra como você pode utilizar Like para testar expressões para diferentes padrões.

Tipo de correspondência

Padrão	Coincidente (retorna True)	Não coincidente (retorna False)
Vários caracteres	a*a	aa, aBa, aBBBa aBC *ab* abc, AABB, Xab aZb, bac
Caractere especial	a[*]a	a*a aaa
Vários caracteres	ab*	abcdefg, abc cab, aab
Um único caractere	a?a	aaa, a3a, aBa aBBBa
Um único dígito	a#a	a0a, a1a, a2a aaa, a10a
Intervalo de caracteres	[a-z]	f, p, j 2, &
Fora de um intervalo	[!a-z]	9, &, % b, a
Nenhum dígito	[!0-9]	A, a, &, ~ 0, 1, 9
Combinado	a[!b-m]#	An9, az0, a99 abc, aj0

10. CARACTERES CURINGA

A correspondência interna de padrões oferece uma ferramenta versátil para fazer comparações de seqüências. A tabela a seguir mostra os caracteres curinga que você pode utilizar com o operador Like e o número de dígitos ou seqüências aos quais eles podem corresponder.

Caractere(s) em padrão	Coincide com expressão
?	Qualquer caractere isolado
*	Zero ou mais caracteres
#	Qualquer dígito isolado (0 — 9)
[listadecaract]	Qualquer caractere isolado em listadecaract
[!listadecaract]	Qualquer caractere isolado não-presente em listadecaract

Você pode utilizar um grupo de um ou mais caracteres (listadecaract) entre colchetes ([ ]) para coincidir com qualquer caractere isolado em expressão, e listadecaract pode incluir praticamente qualquer caractere do conjunto de caracteres ANSI, inclusive dígitos. De fato, você pode utilizar os caracteres especiais colchete de abertura ([ ), ponto de interrogação (?), sinal de número (#) e asterisco (*) para que correspondam diretamente a eles mesmos somente se estiverem entre colchetes. Você pode utilizar o colchete de fechamento ( ]) dentro de um grupo para que corresponda a ele mesmo, mas pode utilizá-lo fora de um grupo como um caractere individual.

Além de uma simples lista de caracteres entre colchetes, listadecaract pode especificar um intervalo de caracteres através da utilização de um hífen (-) para separar os limites superior e inferior do intervalo. Por exemplo, a utilização de [A-Z] em padrão resultará em uma correspondência se a posição do caractere correspondente em expressão contiver qualquer uma das letras maiúsculas no intervalo de A a Z. Você pode incluir vários intervalos entre os colchetes sem delimitar os intervalos. Por exemplo, [a-zA-Z0-9] corresponde a qualquer caractere alfanumérico.

Outras regras importantes para correspondência de padrão incluem:

Um ponto de exclamação (!) no início de listadecaract significa que uma correspondência será feita se qualquer caractere, exceto aqueles na listadecaract, for encontrado em expressão. Quando utilizado sem colchetes, o ponto de exclamação corresponderá a si mesmo.

Você pode utilizar o hífen (-) seja no início (depois de um ponto de exclamação, se este for utilizado) ou no fim de listadecaract para corresponder a si mesmo. Se estiver em qualquer outro local, o hífen identificará um intervalo de caracteres ANSI.

Quando você especifica um intervalo de caracteres, os caracteres devem aparecer em ordem de classificação crescente (A-Z ou 0-100). [A-Z] é um padrão válido, mas [Z-A] não é.

A seqüência de caracteres [ ] é ignorada; ela é considerada uma seqüência de comprimento zero ("").