

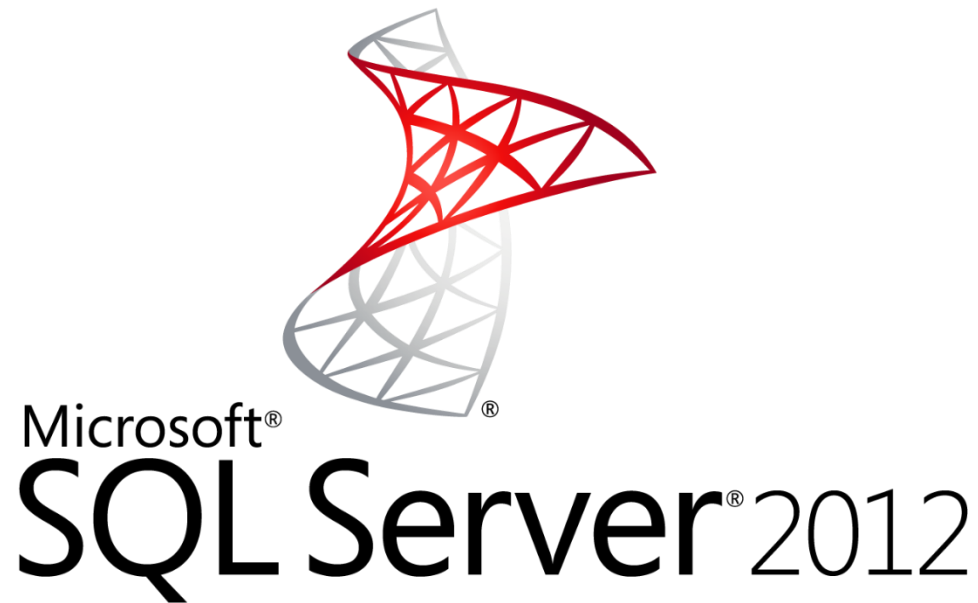


CWI SOFTWARE

módulo 2 | banco de dados

Crescer 2015-1

Feevale



4 - Comandos SQL – Joins

André Luís Nunes

setembro/2015

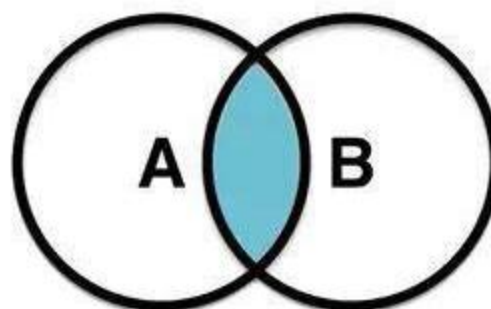
RELACIONAMENTOS: tipos

Para consultar informações de mais de uma tabela é preciso relacioná-las, para isso devemos utilizar o seguinte:

- **Inner Join:** permite relacionar todos os registros de uma tabela comparando com outra tabela.
- **Outer (right) Join:** força o retorno de registros de uma tabela, mesmo que não exista um registro correspondente na outra.
- **Sub-query:** permite relacionar a existência (ou negação) de registros com outra consulta. Normalmente utilizada com EXISTS.
- **Cross join:** produto cartesiano de uma consulta.

RELACIONAMENTO: inner join

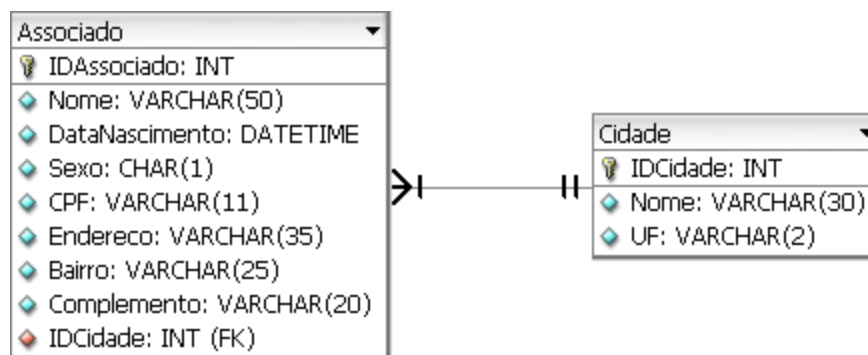
Relacionamento mais comum:



```
SELECT <fields list>  
FROM TableA A  
INNER JOIN TableB B  
ON A.Key = B.Key
```

RELACIONAMENTO: inner join

Relacionamento mais comum.



- » Quando uma consulta é realizada sobre mais de uma tabela, relacionando-os com joins é necessário especificar a qual tabela pertence a coluna.
- » Se o campo possuir o mesmo nome em mais de uma tabela ocorrerá erro se não for informada a tabela.

RELACIONAMENTO: inner join

Sintaxe do comando:

```
Select A.<coluna>, D.<coluna>  
  From <tabelaPrincipal> A  
  Inner Join <tabelaSecundaria> D  
           on A.<IDTabelaD> = D.<IDTabelaD>  
  Where <condicao>  
  Group by <coluna>  
  Having <condicao>  
  Order by <coluna>
```

RELACIONAMENTO: inner join

Recuperando todos os associados que tenham Cidade cadastrada:

```
Select a.Nome as NomeAssociado,  
       c.Nome as NomeCidade  
From Associado a  
      INNER JOIN Cidade c ON c.IDCidade = a.IDCidade
```

Inner join explícito

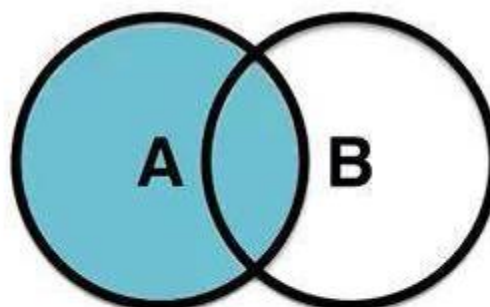
Outra forma de escrever esta consulta é através do Where, as tabelas são adicionadas na cláusula WHERE (separadas por vírgula) e os relacionamentos são escritos no WHERE (“misturados” com demais filtros da pesquisa):

```
Select a.Nome as NomeAssociado,  
       c.Nome as NomeCidade  
From Associado a, Cidade c  
      WHERE c.IDCidade = a.IDCidade
```

Inner join implícito

RELACIONAMENTO: left (outer) join

Recuperando todos os dados de uma fonte, mesmo que não exista em outra (relacionamento opcional):



```
SELECT <fields list>  
FROM TableA A  
LEFT JOIN TableB B  
ON A.Key = B.Key
```


RELACIONAMENTO: left (outer) join

Recuperando todos os dados de uma fonte, mesmo que não exista em outra:

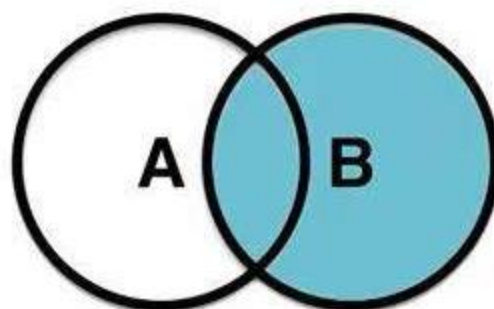
```
Select a.Nome as NomeAssociado,  
       c.Nome as NomeCidade  
From Associado a  
     LEFT JOIN Cidade c ON c.IDCidade = a.IDCidade
```

» Mesmo que exista algum registro na tabela Associado com IDCidade nulo este será exibido:

```
Select a.Nome as NomeAssociado,  
       c.Nome as NomeCidade  
From Associado a, Cidade c  
Where c.IDCidade *= a.IDCidade
```

RELACIONAMENTO: Right (outer) join

Recuperando todos os dados de uma fonte, mesmo que não exista em outra:



```
SELECT <fields list>  
FROM TableA  
RIGHT JOIN TableB B  
ON A.Key = B.Key
```

RELACIONAMENTO: Right (outer) join

Recuperando todos os dados de uma fonte, mesmo que não exista em outra:

```
Select a.Nome as NomeAssociado,  
       c.Nome as NomeCidade  
From Associado a  
      RIGHT JOIN Cidade c ON c.IDCidade = a.IDCidade
```

» Mesmo que exista algum registro na tabela Cidade não tenha registro em Associado:

```
Select a.Nome as NomeAssociado,  
       c.Nome as NomeCidade  
From Associado a, Cidade c  
Where c.IDCidade =* a.IDCidade
```

RELACIONAMENTO: Self join (auto-relacionamento)

Relacionar com a própria tabela.

É necessário adicionar a tabela novamente, da mesma que forma que um INNER JOIN (ou left join conforme a necessidade).

```
Select e.NomeEmpregado as NomeEmpregado,  
       g.NomeEmpregado as NomeGerente  
From Empregado e  
     INNER JOIN Empregado g ON e.IDGerente = g.IDEmpregado
```

» Serão exibidos somente os empregados que tenham gerente.

```
Select e.Nome as NomeEmpregado,  
       g.Nome as NomeGerente  
From Empregado e, Empregado g  
Where e.IDGerente = g.IDEmpregado
```

RELACIONAMENTO: cross join

É o resultado de uma consulta onde as tabelas não são relacionadas. Com isso para cada linha de uma tabela A todos os registros da tabela B serão exibidos.

A		B		Resultado			
Col1	Col2	Col_1	Col_2	Col1	Col2	Col_1	Col_2
ABC	10	ABB	102	ABC	10	ABB	102
BCD	20	ACC	304	BCD	20	ABB	102
		ADD	508	ABC	10	ABB	102
				BCD	20	ACC	304
				ABC	10	ACC	304
				BCD	20	ACC	304

```
Select a.IDAssociado,
       a.Nome,
       a.IDCidade IDCidadeEmp,
       c.IDCidade,
       c.Nome
From Associado a, Cidade c
go
```

ATENÇÃO: NUNCA ESQUEÇA DE RELACIONAR TODAS AS TABELAS EM UMA CONSULTA.

RELACIONAMENTO: sub-queries - exists (com join)

É uma query dentro de outra. Pode ser aplicado nos comandos Insert, Update e Delete também.

No **EXISTS** é necessário **relacionar** a consulta interna (subquery) com a consulta principal:

```
Select IDCidade, Nome  
  From Cidade e  
 Where EXISTS (Select 1  
                From   Associado a  
                Where  a.IDCidade = e.IDCidade)
```

» Exibirá todas as cidades que tenham associado relacionado.

RELACIONAMENTO: sub-queries - sem join

Sub-query sem relacionar com a principal: IN, = (igual), != (diferente), e outros.

» Esta consulta apresenta desempenho inferior ao usado com EXISTS.

```
Select IDCidade, Nome  
  From Cidade e  
 Where IDCidade IN (Select IDCidade  
                   From Associado)
```

» Exibe a cidade de menor IDCidade do estado de SP:

```
Select IDCidade, Nome  
  From Cidade e  
 Where IDCidade = (Select MIN(IDCidade)  
                   From Cidade  
                   Where UF = 'SP')
```

Para utilizar o comparador de igualdade é preciso garantir que retornará somente 1 registro na sub-query.

RELACIONAMENTO: sub-queries - no select

É possível utilizar uma consulta como coluna, na cláusula SELECT.

» Esta consulta exibirá o nome da cidade:

```
Select Nome,  
      (Select Nome  
       From   Cidade c  
       Where  c.IDCidade = a.IDCidade) as NomeCidade  
From Associado a  
go
```

Para utilizar uma sub-query no Select devemos garantir que a consulte não retorne mais de 1 linha, e que apenas 1 (uma) coluna,

RELACIONAMENTO: sub-queries - no from

É possível utilizar uma consulta como fonte de dados, na cláusula FROM.

```
Select a.Nome, city.Nome as NomeCidade
  From Associado a
  Inner join (Select IDCidade, (Nome + '-' + UF) as Nome
             From Cidade
             Where UF in ('SC', 'PR', 'RS') ) as city
    on city.IDCidade = a.IDCidade
go
```

RELACIONAMENTO: union

É a união de duas (ou mais) consultas, obrigatoriamente o número de colunas e tipos devem ser iguais em ambas as consultas. Existe duas formas:

- **UNION:** elimina as linhas duplicadas do resultado final (mais lento);
- **UNION ALL:** não elimina as linhas duplicadas (mais rápido).

A		B		Resultado	
Col1	Col2	Col1	Col2	Col1	Col2
ABC	10	ABB	102	ABC	10
BCD	20	ACC	304	BCD	20
		ADD	508	ABB	102
				ACC	304
				ADD	508

RELACIONAMENTO: union

Retornando em uma consulta todos os nomes das tabelas Empregado e Associado.

```
Select Nome  
From    Associado  
UNION ALL  
Select NomeEmpregado  
From    Empregado  
go
```

RELACIONAMENTOS



mãos à obra

Inserção de registros, execute os comandos abaixo

- Atualizando IDCidade

```
update associado  
set idcidade = 1  
where idassociado = 1;
```

```
update associado  
set idcidade = 32  
where idassociado = 3;
```

RELACIONAMENTOS — exercícios



- 1) Selecione o nome do empregado e nome do departamento que cada um está associado.
- 2) Exibir o nome do associado e sua cidade, exibir também associados sem Cidade relacionada.
- 3) Lista os estados (UF) e total de cidades que não possuem associados relacionados (dica: pode ser utilizado count + group by + exists).
- 4) Faça uma consulta que liste o nome do associado, o nome da cidade, e uma coluna que indique se a cidade é da região SUL (RS, SC, PR), se for imprimir *** (3 asteriscos), senão imprimir nulo.

RELACIONAMENTOS — exercícios



- 5) Liste o nome do empregado, o nome do gerente, e o departamento de cada um.
- 6) Faça uma cópia da tabela Empregado e altere o salário de todos os empregados que o departamento fique na localidade de SAO PAULO, faça um reajuste de 14,5%
- 7) Liste a diferença que representará o reajuste aplicado no item anterior no somatório dos salários de todos os empregados.
- 8) Liste o departamento que possui o empregado de maior salário.

RELACIONAMENTOS — exercícios



- 9) Faça uma consulta para exibir o nome de cada associado e sua cidade e juntamente com os empregados (nome) e a cidade (localidade) de seu departamento, isto deve ser exibido em uma consulta.
- 10) Lista as cidades que tenham associado relacionado.

André Nunes

Núcleo de Tecnologia

andrenunes@cwí.com.br

(51) 3081.3622