



PELO FUTURO DO TRABALHO



05- Seleção de dados com várias tabelas

Professor Hermano Roepke

Contextualização

O conceito que envolve o recurso de busca em vários objetos de banco de dados está atrelado ao termo **junção**. Uma junção (join) em banco de dados é uma condição pela qual os dados são submetidos a outro conjunto de dados (uma tabela, por exemplo).

Existem quatro tipos de junções, mas antes vamos conhecer algumas premissas para o uso das junções e a sintaxe utilizada.

Como utilizar uma junção

Para utilizar uma junção, deve-se relacionar colunas de diferentes tabelas no banco de dados. A forma mais comum de fazer isso é através da seguinte sintaxe:

```
SELECT tabela1.coluna, tabela2.coluna  
FROM tabela1, tabela2  
WHERE tabela1.coluna = tabela2.coluna;
```

Não é uma condição elementar comum a todos os SGBDs, mas boa prática na escrita de comandos é utilizar como prefixo de cada coluna o nome da tabela (ou o apelido, conforme no exemplo abaixo). Isso vai ajudar na legibilidade do comando assim como prevenir possíveis problemas que o SGBD que você estiver utilizando venha a submeter o comando.

```
SELECT t1.coluna, t2.coluna  
FROM tabela1 t1, tabela2 t2  
WHERE t1.coluna = t2.coluna;
```

Exemplo através de um cenário

Vamos criar comandos para exemplificar os conceitos utilizando os dados destas tabelas.

EMPREGADO

ID_EMPREGADO	NM_EMPREGADO	...	ID_DEPARTAMENTO
7839	KING	...	10
7698	BLAKE	...	30
...
7934	MILLER	...	10

Possui 14 linhas
(registros)

DEPARTAMENTO

ID_DEPARTAMENTO	NM_DEPARTAMENTO	LOCALIZACAO
10	CONTÁBIL	NEW YORK
20	PESQUISA E DESENVOLVIMENTO	DALLAS
30	VENDAS	CHICAGO
40	OPERAÇÕES	BOSTON

Possui 4 linhas
(registros)

GRADE_SALARIAL

ID_GRADE	MENOR_SALARIO	MAIORSALARIO
1	700	1200
2	1201	1400
3	1401	2000
4	2001	3000
5	3001	9999

Possui 5 linhas
(registros)

Junção Idêntica

A junção idêntica ocorre quando fazemos uma condição onde uma chave estrangeira (FK) de uma tabela é associada a uma chave primária (PK) de outra tabela. No exemplo a seguir, temos a coluna "ID_DEPARTAMENTO", que é uma FK da tabela "EMPREGADO", sendo comparada com a coluna "ID_DEPARTAMENTO", que é a PK da tabela "DEPARTAMENTO".

EMPREGADO

ID_EMPREGADO	NM_EMPREGADO	ID_DEPARTAMENTO
7369	SMITH	20
7499	ALLEN	30
7521	WARD	30
7566	JONES	20
7654	MARTIN	30
7698	BLAKE	30
7782	CLARK	10
7788	SCOTT	20
7839	KING	10
7844	TURNER	30
7876	ADAMS	20
7900	JAMES	30
7902	FORD	20
7934	MILLER	10

14 linhas selecionadas

DEPARTAMENTO

ID_DEPARTAMEN	NM_DEPARTAMENTO	LOCALIZACAO
10	CONTÁBIL	NEW YORK
10	CONTÁBIL	NEW YORK
10	CONTÁBIL	NEW YORK
20	PESQUISA E DESENVOLVIMENTO	DALLAS
20	PESQUISA E DESENVOLVIMENTO	DALLAS
20	PESQUISA E DESENVOLVIMENTO	DALLAS
20	PESQUISA E DESENVOLVIMENTO	DALLAS
20	PESQUISA E DESENVOLVIMENTO	DALLAS
30	VENDAS	CHICAGO
30	VENDAS	CHICAGO
30	VENDAS	CHICAGO
30	VENDAS	CHICAGO
30	VENDAS	CHICAGO
30	VENDAS	CHICAGO

14 linhas selecionadas

Exemplo da junção idêntica

Imagine que você queira produzir uma consulta cujo resultado apresente as colunas “ID_EMPREGADO”, “ID_DEPARTAMENTO” e “LOCALIZACAO”, conforme visualizado abaixo:

EMPREGADO

ID_EMPREGADO	NM_EMPREGADO	...	ID_DEPARTAMENTO
7839	KING	...	10
7698	BLAKE	...	30
...
7934	MILLER	...	10

DEPARTAMENTO

ID_DEPARTAMENTO	NM_DEPARTAMENTO	LOCALIZACAO
10	CONTÁBIL	NEW YORK
20	PESQUISA E DESENVOLVIMENTO	DALLAS
30	VENDAS	CHICAGO
40	OPERAÇÕES	BOSTON

ID_EMPREGADO	ID_DEPARTAMENTO	LOCALIZACAO
7369	20	DALLAS
7499	30	CHICAGO
7782	10	NEW YORK
7788	20	DALLAS
7839	10	NEW YORK
7844	30	CHICAGO

...

14 linhas recuperadas

```
SELECT e.id_empregado, e.id_departamento, d.localizacao
FROM empregado e, departamento d
WHERE e.id_departamento = d.id_departamento;
```

Chave estrangeira

Chave primária

Comando equivalente

```
SELECT e.id_empregado, e.id_departamento, d.localizacao
FROM empregado e join departamento d on
e.id_departamento = d.id_departamento;
```

O problema do produto cartesiano

Um produto cartesiano é formado quando:

- uma condição de junção estiver omitida;
- uma condição de junção estiver inválida;
- todas as linhas da primeira tabela estiver unidas e todas as linhas da segunda tabela.

EMPREGADO

ID_EMPREGADO	NM_EMPREGADO	...	ID_DEPARTAMENTO
7839	KING	...	10
7698	BLAKE	...	30
...
7934	MILLER	...	10

DEPARTAMENTO

ID_DEPARTAMENTO	NM_DEPARTAMENTO	LOCALIZACAO
10	CONTÁBIL	NEW YORK
20	PESQUISA E DESENVOLVIMENTO	DALLAS
30	VENDAS	CHICAGO
40	OPERAÇÕES	BOSTON

“Produto
Cartesiano:
 $14 \times 4 = 56$ linhas”

NM_EMPREGADO	NM_DEPARTAMENTO
KING	CONTÁBIL
BLAKE	CONTÁBIL
...	...
KING	PESQUISA E DESENVOLVIMENTO
BLAKE	PESQUISA E DESENVOLVIMENTO
...	...

56 linhas recuperadas

Evitando o produto cartesiano

Para evitar um produto cartesiano, sempre inclua uma condição de junção válida em uma cláusula WHERE. Outra dica importante é sempre observar a quantidade de tabelas adicionadas na cláusula FROM. Há uma regra simples que vale a pena seguir: o número de tabelas na cláusula FROM menos um (n° tabelas - 1) de junções válidas na cláusula WHERE. Vamos a um exemplo. Abaixo temos nosso cenário com a sentença SQL gerando um produto cartesiano:

```
SELECT e.id_empregado, e.id_departamento, d.localizacao  
FROM empregado e, departamento d;
```

Você percebeu que não há cláusula WHERE; consequentemente, nenhuma condição de junção foi criada. No entanto, observe que há duas tabelas declaradas na cláusula FROM. Portanto, ao aplicarmos nossa regra, no mínimo uma junção válida deveria existir.

Obs: A chave estrangeira de uma tabela é composta por duas ou mais colunas, cada coluna deverá apresentar uma junção para que o produto cartesiano seja evitado.

Condições adicionais de pesquisa

Aliada à junção, você pode acrescentar uma condição(ões) adicional(is) para filtrar o resultado. Considere o exemplo abaixo, onde se pretende recuperar as informações do empregado “KING”.

EMPREGADO

ID_EMPREGADO	NM_EMPREGADO	ID_DEPARTAMENTO
7782	CLARK	10
7839	KING	10
7934	MILLER	10
7369	SMITH	20
7566	JONES	20
7788	SCOTT	20
7876	ADAMS	20
7902	FORD	20
7499	ALLEN	30
7521	WARD	30
7654	MARTIN	30
7698	BLAKE	30
7844	TURNER	30
7900	JAMES	30

...

14 linhas selecionadas

DEPARTAMENTO

ID_DEPARTAMEN	NM_DEPARTAMENTO	LOCALIZACAO
10	CONTÁBIL	NEW YORK
10	CONTÁBIL	NEW YORK
10	CONTÁBIL	NEW YORK
20	PESQUISA E DESENVOLVIMENTO	DALLAS
20	PESQUISA E DESENVOLVIMENTO	DALLAS
20	PESQUISA E DESENVOLVIMENTO	DALLAS
20	PESQUISA E DESENVOLVIMENTO	DALLAS
20	PESQUISA E DESENVOLVIMENTO	DALLAS
20	PESQUISA E DESENVOLVIMENTO	DALLAS
30	VENDAS	CHICAGO
30	VENDAS	CHICAGO
30	VENDAS	CHICAGO
30	VENDAS	CHICAGO
30	VENDAS	CHICAGO
30	VENDAS	CHICAGO

...

14 linhas selecionadas

Exemplo de junção idêntica condicionada

Imagine que você queira produzir uma consulta cujo resultado apresente as colunas “ID_EMPREGADO”, “ID_DEPARTAMENTO” e “LOCALIZACAO”, conforme visualizado abaixo:

```
SELECT e.id_empregado, e.id_departamento, d.localizacao  
FROM empregado e, departamento d  
WHERE e.id_departamento = d.id_departamento AND  
e.nm_empregado = 'KING';
```

Resultado:

	id_empregado	id_departamento	localizacao
▶	7839	10	NEW YORK

Junção externa

Use uma junção externa para consultar também todas as linhas que em geral não atendem à condição de junção. No exemplo abaixo, podemos visualizar um cenário onde o departamento 40 (OPERAÇÕES) não está vinculado a nenhum empregado.

EMPREGADO			DEPARTAMENTO	
NM_EMPREGADO	...	ID_DEPARTAMENTO	ID_DEPARTAMENTO	NM_DEPARTAMENTO
KING	...	10	10	CONTÁBIL
BLAKE	...	30	30	VENDAS
CLARK	...	10	10	CONTÁBIL
JONES	...	20	20	PESQUISA E DESENVOLVIMENTO
...			...	
			40	OPERAÇÕES



Nenhum funcionário do departamento OPERAÇÕES

Se você construir um comando SELECT utilizando uma junção idêntica, a linha (registro) que não estiver relacionada estará ausente no resultado. E é exatamente esse papel da junção externa, ou seja, permitir que todas as linhas sejam apresentadas, independente da condição de relação.

Sintaxe da junção externa

O padrão ANSI apresenta duas formas de utilização da junção externa. Veja:

```
SELECT tabela1.coluna1, tabela2.coluna1  
FROM tabela1 LEFT JOIN tabela2 ON  
(tabela1.coluna1 = tabela2.coluna1);
```

```
SELECT tabela1.coluna1, tabela2.coluna1  
FROM tabela1 RIGHT JOIN tabela2 ON  
(tabela1.coluna1 = tabela2.coluna1);
```

No caso do SGBD Oracle como alternativa é possível realizar uma junção externa através da simbologia “(+)” ao lado da coluna observe:

```
SELECT tabela1.coluna1, tabela2.coluna1  
FROM tabela1, tabela2  
(tabela1.coluna1(+) = tabela2.coluna1);
```

Exemplo de junção externa

Considerando o mesmo cenário o conjunto de dados, temos:

```
SELECT e.nm_empregado, e.id_departamento, d.nm_departamento, d.localizacao  
FROM empregado e LEFT JOIN departamento d ON (e.id_departamento = d.id_departamento);
```

```
SELECT e.nm_empregado, e.id_departamento, d.nm_departamento, d.localizacao  
FROM empregado e RIGHT JOIN departamento d ON (e.id_departamento = d.id_departamento);
```

Junção não idêntica

É possível realizar uma junção entre duas ou mais tabelas sem que exista uma chave estrangeira estabelecendo a relação/ligação entre elas. Essa junção é conhecida como **junção não idêntica**. Nesse caso, o relacionamento é obtido por meio de um operador que não o igual “=”. Considere o seguinte cenário.

EMPREGADO		
ID_EMPREGADO	NM_EMPREGADO	SALARIO
7839	KING	5000
7698	BLAKE	2850
7782	CLARK	2450
7566	JONES	2975
7654	MARTIN	1250
7499	ALLEN	1600
7844	TURNER	1500
7900	JAMES	950
...		

14 linhas selecionadas

GRADE_SALARIAL		
ID_GRADE	MENOR_SALARIO	MAIORSALARIO
1	700	1200
2	1201	1400
3	1401	2000
4	2001	3000
5	3001	9999



O salário na tabela EMPREGADO está entre o menor salário e maior salário na tabela GRADE_SALARIAL

Exemplo de junção não idêntica

EMPREGADO

ID_EMPREGADO	NM_EMPREGADO	SALARIO
7839	KING	5000
7698	BLAKE	2850
7782	CLARK	2450
7566	JONES	2975
7654	MARTIN	1250
7499	ALLEN	1600
7844	TURNER	1500
7900	JAMES	950
...		

14 linhas selecionadas

GRADE_SALARIAL

ID_GRADE	MENOR_SALARIO	MAIORSALARIO
1	700	1200
2	1201	1400
3	1401	2000
4	2001	3000
5	3001	9999



O salário na tabela EMPREGADO está entre o menor salário e maior salário na tabela GRADE_SALARIAL

```
SELECT e.id_empregado, e.nm_empregado, g.id_grade
FROM empregado e, grade_salarial g
WHERE e.salario BETWEEN g.menor_salario and g.maior_salario;
```



Comando equivalente

```
SELECT e.id_empregado, e.nm_empregado, g.id_grade
FROM empregado e, grade_salarial g
WHERE e.salario >= g.menor_salario and e.salario <= g.maior_salario;
```


Auto junção

Você já parou para analisar como produzir resultados utilizando estruturas que apresentam auto relacionamento ou relacionamento recursivo binário? A solução é simples utilizar a **auto junção**. Assim como no relacionamento, onde uma tabela exerce o papel de duas (distintas), o mesmo ocorre na auto junção. Considere o cenário Abaixo:

EMPREGADO (EMPREGADO)			EMPREGADO (LIDER)	
ID_EMPREGADO	NM_EMPREGADO	LIDER	ID_EMPREGADO	NM_EMPREGADO
7839	KING			
7698	BLAKE	7839	7839	KING
7782	CLARK	7839	7839	KING
7566	JONES	7839	7839	KING
7654	MARTIN	7698	7698	BLAKE
7499	ALLEN	7698	7698	BLAKE


LIDER na tabela EMPREGADO é igual a ID_EMPREGADO na tabela LIDER

Observe que a coluna “LIDER” apresenta valores que correspondem à coluna “ID_EMPREGADO” da própria tabela. É importante ressaltar ainda que a coluna “LIDER” é uma chave estrangeira para a própria tabela –característica esta peculiar do auto relacionamento.

Exemplo de auto junção

Vamos construir uma sentença SQL para recuperar o nome do empregado e o nome do LÍDER (que também é um empregado). Para isso, vamos utilizar a tabela “EMPREGADO” exercendo o papel do empregado e, ao mesmo tempo, exercendo o papel do LÍDER. Veja novamente o cenário e a sentença SQL construída:

EMPREGADO (EMPREGADO)			EMPREGADO (LIDER)	
ID_EMPREGADO	NM_EMPREGADO	LIDER	ID_EMPREGADO	NM_EMPREGADO
7839	KING			
7698	BLAKE	7839	7839	KING
7782	CLARK	7839	7839	KING
7566	JONES	7839	7839	KING
7654	MARTIN	7698	7698	BLAKE
7499	ALLEN	7698	7698	BLAKE



```
SELECT e.nm_empregado, l.nm_empregado LIDER
FROM empregado e, empregado l
WHERE e.lider = l.id_empregado;
```

Resumo

Você vai descobrir em breve que utilizar junções é algo muito comum na construção de sentenças SQL.

Teve também a oportunidade de conhecer as características, aplicações e exemplos de utilização dos quatro tipos de junções (idêntica, externa, não idêntica e auto junção).



PELO FUTURO DO TRABALHO

sesisc.org.br     **sc.senai.br**

0800 048 1212

Rodovia Admar Gonzaga, 2765 - Itacorubi - 88034-001 - Florianópolis, SC