

Consultas tipo Join

Tendo como base a Wikipedia ([http://en.wikipedia.org/wiki/Join_\(SQL\)](http://en.wikipedia.org/wiki/Join_(SQL)))

A cláusula **JOIN** em SQL combina registros de duas tabelas e resulta em uma nova tabela (temporária), também chamada de joined table. O SQL:2003 especifica dois tipos de JOIN: o inner e o outer.

Matematicamente um JOIN consiste em uma função de composição.

Tabelas de Exemplo:

department

DepartmentID	DepartmentName
31	Sales
33	Engineering
34	Clerical
35	Marketing

employee

LastName	DepartmentID
Rafferty	31
Jones	33
Steinberg	33
Robinson	34
Smith	34
Jasper	36

INNER JOIN

Este JOIN combina cada registro de A com cada registro de B respeitando o predicado do JOIN. O resultado será a combinação de todos os registros de A com os de B. Este é o tipo mais comum de JOIN e é o tipo default.

A cláusula JOIN pode vir explícita ou implícita.

Muito cuidado ao unir tabelas que contenham registros com valor NULL, nestes casos devemos usar no predicado IS NULL ou NOT IS NULL para garantir o resultado esperado.

Exemplo de um INNER JOIN explícito aplicado nas duas tabelas anteriores:

```
SELECT *
FROM   employee
       INNER JOIN department
       ON employee.DepartmentID = department.DepartmentID
```

Exemplo com INNER JOIN implícito:

```
SELECT *
FROM   employee, department
```

```
WHERE employee.DepartmentID = department.DepartmentID
```

Ambos irão resultar em:

Employee.LastName	Employee.DepartmentID	Department.DepartmentName	Department.DepartmentID
Smith	34	Clerical	34
Jones	33	Engineering	33
Robinson	34	Clerical	34
Steinberg	33	Engineering	33
Rafferty	31	Sales	31

Obs.: veja que o empregado Jasper e o departamento Marketink não aparecem na tabela resultante. Isso ocorre por que eles não atendem ao predicado WHERE employee.DepartmentID = department.DepartmentID.

Tipos de INNER JOIN

Equi-join – é um tipo que utiliza apenas o sinal de igualdade na condição (predicado).

O mesmo exemplo anterior agora usando EQUI-JOIN:

```
SELECT *
FROM   employee
       INNER JOIN department
         ON employee.DepartmentID = department.DepartmentID
```

Alguns SGBDs tem uma variante para este tipo, veja a sintaxe respectiva para detalhes.

Natural-Join – é uma especialização do Equi-join. Consiste em ter na condição (predicado) uma comparação de todos os campos de ambas as tabelas que tem o mesmo campo na tabela resultante.

Aplicando ao exemplo anterior:

```
SELECT *
FROM   employee NATURAL JOIN department
```

O resultado será:

Employee.LastName	DepartmentID	Department.DepartmentName
Smith	34	Clerical
Jones	33	Engineering
Robinson	34	Clerical
Steinberg	33	Engineering
Rafferty	31	Sales

Por conta de somente o campo DepartamentoID ser comum em ambas as tabelas.

Cross-join – ou Join cartesiano. Representa. Para A e B, então Cross-join = AXB. É similar ao Inner-join onde a cláusula Where foi avaliada como TRUE.

Exemplo de CROSS-JOIN explícito:

```
SELECT *  
FROM employee CROSS JOIN department;
```

Exemplo de CROSS-JOIN implícito:

```
SELECT *  
FROM employee, department;
```

Que resultará:

Employee.LastName	Employee.DepartmentID	Department.DepartmentName	Department.DepartmentID
Rafferty	31	Sales	31
Jones	33	Sales	31
Steinberg	33	Sales	31
Smith	34	Sales	31
Robinson	34	Sales	31
Jasper	36	Sales	31
Rafferty	31	Engineering	33
Jones	33	Engineering	33
Steinberg	33	Engineering	33
Smith	34	Engineering	33
Robinson	34	Engineering	33
Jasper	36	Engineering	33
Rafferty	31	Clerical	34
Jones	33	Clerical	34
Steinberg	33	Clerical	34
Smith	34	Clerical	34
Robinson	34	Clerical	34
Jasper	36	Clerical	34
Rafferty	31	Marketing	35
Jones	33	Marketing	35
Steinberg	33	Marketing	35
Smith	34	Marketing	35
Robinson	34	Marketing	35
Jasper	36	Marketing	35

CROSS-JOIN não aplica nenhuma condição, nenhum predicado.

Outer-join – não requer que cada registro nas duas tabelas unidas tenha um registro comum na outra tabela. Cada registro é retido na tabela resultante mesmo sem existir seu par.

Outer-join subdivide-se em left outer joins, right outer joins e full outer joins.

Para uma tabela ser qualificada como Left ou Right seu nome deve aparecer após a palavra FROM ou JOIN.

Left Outer Join – Para tabelas A e B, o resultado conterá todos os registros de A (left), independente da condição e somente os de B que satisfaçam a condição. Se a condição não trouxer alguns registros de B, então a resultante preencherá com NULL os registros de B sem valor.

Exemplo de Left Outer Join:

```
SELECT *
FROM   employee
       LEFT OUTER JOIN department
         ON employee.DepartmentID = department.DepartmentID
```

Employee.LastName	Employee.DepartmentID	Department.DepartmentName	Department.DepartmentID
Jones	33	Engineering	33
Rafferty	31	Sales	31
Robinson	34	Clerical	34
Smith	34	Clerical	34
Jasper	36	NULL	NULL
Steinberg	33	Engineering	33

Right Outer Join – Semelhante ao Left, sendo que trará todos os registros de B (right), independente da condição e os de A somente atendendo à condição.

Exemplo de Right Outer Join:

```
SELECT *
FROM   employee
       RIGHT OUTER JOIN department
         ON employee.DepartmentID = department.DepartmentID
```

Employee.LastName	Employee.DepartmentID	Department.DepartmentName	Department.DepartmentID
Smith	34	Clerical	34
Jones	33	Engineering	33
Robinson	34	Clerical	34
Steinberg	33	Engineering	33
Rafferty	31	Sales	31
NULL	NULL	Marketing	35

Full Outer Join – Combina os resultados de left e right outer joins. A tabela resultante deve conter todos os registros de ambas as tabelas e preencher com NULL aqueles sem similares em cada lado.

Exemplo de full outer join:

```
SELECT *  
FROM   employee  
       FULL OUTER JOIN department  
         ON employee.DepartmentID = department.DepartmentID
```

Employee.LastName	Employee.DepartmentID	Department.DepartmentName	Department.DepartmentID
Smith	34	Clerical	34
Jones	33	Engineering	33
Robinson	34	Clerical	34
Jasper	36	NULL	NULL
Steinberg	33	Engineering	33
Rafferty	31	Sales	31
NULL	NULL	Marketing	35

Alguns SGBDs não suportam mas podem emular esta funcionalidade.

Para mais detalhes sobre JOIN veja o link da Wikipédia [http://en.wikipedia.org/wiki/Join_\(SQL\)](http://en.wikipedia.org/wiki/Join_(SQL))