

Introdução a Banco de dados

Comandos SQL Server – Parte 7

INNER JOIN – Consultar dados em duas ou mais Tabelas no SQL Server

JOINS

A cláusula JOIN é usada para combinar dados provenientes de duas ou mais tabelas do banco de dados, baseado em um relacionamento entre colunas destas tabelas. há duas categorias principais de joins:

INNER JOIN – Consultar dados em duas ou mais Tabelas no SQL Server

- ◎ **INNER JOIN:** Retorna linhas (registros) quando houver pelo menos uma correspondência em ambas as tabelas.
- ◎ **OUTER JOIN:** Retorna linhas (registros) mesmo quando não houver ao menos uma correspondência em uma das tabelas (ou ambas). O OUTER JOIN divide-se em LEFT JOIN, RIGHT JOIN e FULL JOIN.

INNER JOIN – Consultar dados em duas ou mais Tabelas no SQL Server

```
SELECT colunas  
FROM tabela1  
INNER JOIN tabela2  
ON tabela1.coluna=tabela2.coluna;
```

INNER JOIN – Consultar dados em duas ou mais Tabelas no SQL Server

Onde **tabela1.coluna** é o nome da primeira tabela concatenado com um ponto e com o nome da coluna chave primária ou estrangeira da tabela, e **tabela2.coluna** é o nome da segunda tabela concatenado com um ponto e com a chave estrangeira ou primária dessa tabela que se relaciona com a chave da primeira tabela.

INNER JOIN – Consultar dados em duas ou mais Tabelas no SQL Server

Vejamos um exemplo. Vamos consultar as tabelas de livros e autores (tbl_livro e tbl_autores) e retornar os dados relativos aos livros e aos autores ao mesmo tempo:

INNER JOIN – Consultar dados em duas ou mais Tabelas no SQL Server

```
SELECT * FROM tbl_Livro
```

```
INNER JOIN tbl_autores
```

```
ON tbl_Livro.ID_Autor = tbl_autores.ID_Autor;
```

	ID_Livro	Nome_Livro	ISBN13	ISBN10	ID_Categoria	ID_Autor	ID_Editora	Data_Pub	Preco_Livro	ID_Autor	Nome_
▶	1	Linux Command Line and Shell Scripting	9781118983843	111898384X	1	5	4	2015-01-09	68.35	5	Richard
	2	SSH, o Shell Seguro	9780596008956	0596008953	1	1	2	2005-05-17	58.30	1	Daniel
	3	Using Samba	9780596002565	0596002564	1	2	2	2003-12-21	61.45	2	Gerald
	4	Fedora and Red Hat Linux	9780133477436	0133477436	1	3	1	2014-01-10	62.24	3	Mark
	5	Windows Server 2012 Inside Out	9780735666313	0735666318	1	4	3	2013-01-25	66.80	4	William
	6	Microsoft Exchange Server 2010	9780735640610	0735640610	1	4	3	2010-12-01	47.20	4	William
	7	Practical Electronics for Inventors	9781259587542	1259587541	1	13	5	2016-03-24	67.80	13	Simon

INNER JOIN – Consultar dados em duas ou mais Tabelas no SQL Server

Vejamos outro exemplo. Faremos a consulta nas tabelas de livros e autores, porém retornando apenas os nomes dos livros, seus ISBNs e os nomes dos autores desses livros. Note que as duas primeiras informações estão na tabela de livros, e o nome do autor está somente na tabela de autores. O que as liga é o relacionamento entre as colunas ID_Autor existentes nas tabelas – é chave primária na tabela de autores e estrangeira na de livros:

INNER JOIN – Consultar dados em duas ou mais Tabelas no SQL Server

```
SELECT tbl_Livro.Nome_Livro, tbl_Livro.ISBN13,  
tbl_autores.Nome_Autor  
FROM tbl_Livro  
INNER JOIN tbl_autores  
ON tbl_Livro.ID_Autor = tbl_autores.ID_Autor;
```

INNER JOIN – Consultar dados em duas ou mais Tabelas no SQL Server

Nome_Livro	ISBN13	Nome_Autor
Linux Command Line and Shell Scripting	9781118983843	Richard
SSH, o Shell Seguro	9780596008956	Daniel
Using Samba	9780596002565	Gerald
Fedora and Red Hat Linux	9780133477436	Mark
Windows Server 2012 Inside Out	9780735666313	William
Microsoft Exchange Server 2010	9780735640610	William
Practical Electronics for Inventors	9781259587542	Simon

INNER JOIN – Consultar dados em duas ou mais Tabelas no SQL Server

Mais um exemplo, agora com filtros. Vamos retornar os nomes dos livros e nomes das editoras, mas somente das editoras cujo nome se inicia com a letra M. Note o uso de alias nestas declarações, a fim de simplificar o código:

INNER JOIN – Consultar dados em duas ou mais Tabelas no SQL Server

```
SELECT L.Nome_Livro AS Livros, E.Nome_editora  
AS Editoras  
FROM tbl_Livro AS L  
INNER JOIN tbl_editoras AS E  
ON L.ID_editora = E.ID_editora  
WHERE E.Nome_Editora LIKE 'M%';
```

	Livros	Editoras
▶	Windows Server 2012 Inside Out	Microsoft Press
	Microsoft Exchange Server 2010	Microsoft Press
	Practical Electronics for Inventors	McGraw-Hill Education

INNER JOIN – Consultar dados em duas ou mais Tabelas no SQL Server

Exemplo sem utilização de Alias para as tabelas

```
SELECT tbl_livro.Nome_Livro AS Livro,  
tbl_editoras.Nome_Editora AS Editora  
FROM tbl_livro  
INNER JOIN tbl_editoras  
ON tbl_livro.ID_Editora=tbl_editoras.ID_Editora  
WHERE tbl_editoras.Nome_Editora LIKE 'M%'
```

	Livros	Editoras
▶	Windows Server 2012 Inside Out	Microsoft Press
	Microsoft Exchange Server 2010	Microsoft Press
	Practical Electronics for Inventors	McGraw-Hill Education

INNER JOIN – Consultar dados em duas ou mais Tabelas no SQL Server

Mais um exemplo para terminar. Agora vamos fazer um INNER JOIN com as três tabelas do banco de dados simultaneamente. Queremos os nomes e preços dos livros, nomes de seus autores e editoras, mas somente das editoras cujo nome se inicia com a letra O, tudo isso ordenado em ordem decrescente de preço dos livros:

INNER JOIN – Consultar dados em duas ou mais Tabelas no SQL Server

```
SELECT L.Nome_Livro AS Livro,  
A.Nome_autor AS Autor,  
E.Nome_Editora AS Editora,  
L.Preco_Livro AS 'Preço do Livro'  
FROM tbl_Livro AS L  
INNER JOIN tbl_autores AS A  
ON L.ID_autor = A.ID_autor  
INNER JOIN tbl_editoras AS E  
ON L.ID_editora = E.ID_editora  
WHERE E.Nome_Editora LIKE 'O%'  
ORDER BY L.Preco_Livro DESC;
```

	Livro	Autor	Editora	Preço do Livro
►	Using Samba	Gerald	O'Reilly	61.45
	SSH, o Shell Seguro	Daniel	O'Reilly	58.30

INNER JOIN – Consultar dados em duas ou mais Tabelas no SQL Server

```
SELECT tbl_livro.Nome_Livro AS Livro,  
tbl_autores.Nome_Autor AS Autor,  
tbl_editoras.Nome_Editora AS Editora,  
tbl_livro.Preco_livro AS 'Preço do Livro'  
FROM tbl_livro  
INNER JOIN tbl_autores  
ON tbl_livro.ID_Autor=tbl_autores.ID_Autor  
INNER JOIN tbl_editoras  
ON tbl_livro.ID_Editora=tbl_editoras.ID_Editora  
--WHERE tbl_editoras.Nome_Editora LIKE 'O%'  
ORDER BY tbl_livro.Preco_livro DESC
```

	Livro	Autor	Editora	Preço do Livro
►	Using Samba	Gerald	O'Reilly	61.45
	SSH, o Shell Seguro	Daniel	O'Reilly	58.30

SQL Server – LEFT e RIGHT JOIN – Consultar dados em duas ou mais tabelas

- ① **LEFT JOIN:** Retorna todas as linhas da tabela à esquerda, mesmo se não houver nenhuma correspondência na tabela à direita.
- ② **RIGHT JOIN:** Retorna todas as linhas da tabela à direita, mesmo se não houver nenhuma correspondência na tabela à esquerda.

SQL Server – LEFT e RIGHT JOIN – Consultar dados em duas ou mais tabelas

Sintaxe:

```
SELECT colunas  
FROM tabela_esq  
LEFT (OUTER) JOIN tabela_dir  
ON tabela_esq.coluna=tabela_dir.coluna;
```

SQL Server – LEFT e RIGHT JOIN – Consultar dados em duas ou mais tabelas

Exemplo:

SELECT * FROM tbl_autores

LEFT JOIN tbl_Livro

ON tbl_Livro.ID_Autor = tbl_autores.ID_Autor;

	ID_Autor	Nome_Autor	Sobrenome_Autor	ID_Livro	Nome_Livro	ISBN13	ISBN10	ID_Categoria	ID_Autor	ID_Editora	
▶	1	Daniel	Barret	2	SSH, o Shell Seguro	9780596008956	0596008953	1	1	2	2
	2	Gerald	Barret	3	Using Samba	9780596002565	0596002564	1	2	2	2
	3	Mark	Sobell	4	Fedora and Red Hat Linux	9780133477436	0133477436	1	3	1	2
	4	William	Stanek	5	Windows Server 2012 Inside Out	9780735666313	0735666318	1	4	3	2
	4	William	Stanek	6	Microsoft Exchange Server 2010	9780735640610	0735640610	1	4	3	2
	5	Richard	Blum	1	Linux Command Line and Shell Scripting	9781118983843	111898384X	1	5	4	2

SQL Server – LEFT e RIGHT JOIN – Consultar dados em duas ou mais tabelas

LEFT JOIN excluindo correspondências:

Sintaxe:

SELECT coluna

FROM tabela_esq

LEFT (OUTER) JOIN tabela_dir

ON tabela_esq.coluna=tabela_dir.coluna

WHERE tabela_dir.coluna IS NULL;

SQL Server – LEFT e RIGHT JOIN – Consultar dados em duas ou mais tabelas

Exemplo:

SELECT * FROM tbl_autores

LEFT JOIN tbl_Livro

ON tbl_Livro.ID_Autor = tbl_autores.ID_Autor

WHERE tbl_Livro.ID_Autor IS NULL;

	ID_Autor	Nome_Autor	Sobrenome_Autor	ID_Livro	Nome_Livro	ISBN13	ISBN10	ID_Categoria	ID_Autor	ID_Editora	Data_Pub	Preco_Livro
▶	6	Jostein	Gaarder	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
	7	Umberto	Eco	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
	8	Neil	De Grasse Tyson	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
	9	Stephen	Hawking	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
	10	Stephen	Jay Gould	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
	11	Charles	Darwin	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
	12	Alan	Turing	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
	14	Paul	Scherz	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

SQL Server – LEFT e RIGHT JOIN – Consultar dados em duas ou mais tabelas

RIGHT JOIN

Sintaxe:

SELECT colunas

FROM tabela_esq

RIGHT JOIN tabela_dir

ON tabela_esq.coluna=tabela_dir.coluna;

SQL Server – LEFT e RIGHT JOIN – Consultar dados em duas ou mais tabelas

Exemplo:

```
SELECT * FROM tbl_Livro AS Li
RIGHT JOIN tbl_editoras AS Ed
ON Li.ID_editora = Ed.ID_editora;
```

	ID_Livro	Nome_Livro	ISBN13	ISBN10	ID_Categoria	ID_Autor	ID_Editora	Data_Pub	Preco_Livro	ID_Editora	Nome_
▶	1	Linux Command Line and Shell Scripting	9781118983843	111898384X	1	5	4	2015-01-09	68.35	4	Wiley
	2	SSH, o Shell Seguro	9780596008956	0596008953	1	1	2	2005-05-17	58.30	2	O'Reilly
	3	Using Samba	9780596002565	0596002564	1	2	2	2003-12-21	61.45	2	O'Reilly
	4	Fedora and Red Hat Linux	9780133477436	0133477436	1	3	1	2014-01-10	62.24	1	Prentice
	5	Windows Server 2012 Inside Out	9780735666313	0735666318	1	4	3	2013-01-25	66.80	3	Microso
	6	Microsoft Exchange Server 2010	9780735640610	0735640610	1	4	3	2010-12-01	47.20	3	Microso
	7	Practical Electronics for Inventors	9781259587542	1259587541	1	13	5	2016-03-24	67.80	5	McGraw

SQL Server – LEFT e RIGHT JOIN – Consultar dados em duas ou mais tabelas

Exemplo:

SELECT * FROM tbl_Livro

RIGHT JOIN tbl_editoras

ON tbl_Livro.ID_editora = tbl_editoras.ID_editora;

	ID_Livro	Nome_Livro	ISBN13	ISBN10	ID_Categoria	ID_Autor	ID_Editora	Data_Pub	Preco_Livro	ID_Editora	Nome_
▶	1	Linux Command Line and Shell Scripting	9781118983843	111898384X	1	5	4	2015-01-09	68.35	4	Wiley
	2	SSH, o Shell Seguro	9780596008956	0596008953	1	1	2	2005-05-17	58.30	2	O'Reilly
	3	Using Samba	9780596002565	0596002564	1	2	2	2003-12-21	61.45	2	O'Reilly
	4	Fedora and Red Hat Linux	9780133477436	0133477436	1	3	1	2014-01-10	62.24	1	Prentice
	5	Windows Server 2012 Inside Out	9780735666313	0735666318	1	4	3	2013-01-25	66.80	3	Microso
	6	Microsoft Exchange Server 2010	9780735640610	0735640610	1	4	3	2010-12-01	47.20	3	Microso
	7	Practical Electronics for Inventors	9781259587542	1259587541	1	13	5	2016-03-24	67.80	5	McGraw

SQL Server – LEFT e RIGHT JOIN – Consultar dados em duas ou mais tabelas

RIGHT JOIN excluindo correspondências:

Sintaxe:

SELECT coluna

FROM tabela_esq

RIGHT JOIN tabela_dir

ON tabela_esq.coluna=tabela_dir.coluna

WHERE tabela_esq.coluna IS NULL;

SQL Server – LEFT e RIGHT JOIN – Consultar dados em duas ou mais tabelas

Exemplo:

```
SELECT * FROM tbl_Livro
RIGHT JOIN tbl_editoras
ON tbl_Livro.ID_editora = tbl_editoras.ID_editora
WHERE tbl_Livro.ID_editora IS NULL;
```

ID_Livro	Nome_Livro	ISBN13	ISBN10	ID_Categoria	ID_Autor	ID_Editora	Data_Pub	Preco_Livro	ID_Editora	Nome_Editora
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	6	Apress
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	7	Bookman
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	8	Bookboon
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	9	Packtpub
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	10	Springer
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	11	Érica
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	12	For Dummies
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	13	Novatec

Operador **UNION** em SQL Server

Assim como uma **JOIN**, uma operação **UNION** permite combinar dados provenientes de duas ou mais tabelas (ou da mesma tabela, com condições diferentes). Porém, em vez de combinar as colunas dessas tabelas, a **UNION** combina as linhas de dois ou mais conjuntos de resultados. Imagine que um **INNER JOIN** é uma operação de intersecção entre conjuntos, e que o **UNION** é uma operação de soma de conjuntos.

Operador **UNION** em SQL Server

Assim, uma **UNION** combina (“une”) duas ou mais declarações **SELECT**. O resultado de cada **SELECT** deve possuir o mesmo número de colunas, e o tipo de dado de cada coluna correspondente deve ser compatível.

Operador UNION em SQL Server

Para ordenar o resultado de uma operação UNION, podemos usar uma declaração **ORDER BY** após o último SELECT codificado. Neste ORDER BY somente é possível usar os nomes de colunas que foram declarados no primeiro SELECT da declaração UNION completa. Os nomes de colunas no resultado final são tirados deste primeiro SELECT.

Operador UNION em SQL Server

Sintaxe:

```
SELECT declaração_1  
UNION [ALL}  
SELECT declaração_2  
UNION [ALL}  
SELECT declaração_3 ...  
[ORDER BY colunas]
```

Operador UNION em SQL Server

Exemplo 01

- Retornar nomes de livros, preços e assuntos dos livros.
- Caso o assunto seja Tecnologia, mostrar o preço acrescido de 15% em seu valor.
- Caso o livro custe mais de 60 reais, descontar 10% em seu valor.
- Ordenar por preço, do mais caro para o mais barato.

Operador UNION em SQL Server

```
SELECT L.Nome_Livro Livro, L.Preco_Livro 'Preço Normal', L.Preco_Livro * 0.90  
'Preço Ajustado', C.Categoria  
FROM tbl_Livro L INNER JOIN tbl_categorias C  
ON L.ID_Categoria = C.ID_Categoria  
WHERE L.Preco_Livro > 60  
UNION  
SELECT L.Nome_Livro Livro, L.Preco_Livro 'Preço Normal', L.Preco_Livro * 1.15  
'Preço Ajustado', C.Categoria  
FROM tbl_Livro L INNER JOIN tbl_categorias C  
ON L.ID_Categoria = C.ID_Categoria  
WHERE C.Categoria = 'Tecnologia'  
ORDER BY 'Preço Ajustado' DESC;
```


Operador UNION em SQL Server

```
SELECT tbl_Livro .Nome_Livro Livro, tbl_Livro .Preco_Livro 'Preço Normal', tbl_Livro
.Preco_Livro * 0.90 'Preço Ajustado', tbl_categorias.Categoria
FROM tbl_Livro
INNER JOIN tbl_categorias
ON tbl_Livro .ID_Categoria = tbl_categorias.ID_Categoria
WHERE tbl_Livro .Preco_Livro > 60
UNION
SELECT tbl_Livro.Nome_Livro Livro, tbl_Livro.Preco_Livro 'Preço Normal',
tbl_Livro.Preco_Livro * 1.15 'Preço Ajustado', tbl_categorias.Categoria
FROM tbl_Livro
INNER JOIN tbl_categorias
ON tbl_Livro.ID_Categoria = tbl_categorias.ID_Categoria
WHERE tbl_categorias.Categoria = 'Tecnologia'
ORDER BY 'Preço Ajustado' DESC;
```

Operador UNION em SQL Server

```
SELECT L.Nome_Livro Livro, L.Preco_Livro 'Preço Normal', L.Preco_Livro * 0.90  
'Preço Ajustado', E.ID_editora  
FROM tbl_Livro L INNER JOIN tbl_editoras E  
ON L.ID_Editora = E.ID_Editora  
WHERE L.Preco_Livro > 60  
UNION  
SELECT L.Nome_Livro Livro, L.Preco_Livro 'Preço Normal', L.Preco_Livro * 1.15  
'Preço Ajustado', E.ID_Editora  
FROM tbl_Livro L INNER JOIN tbl_editoras E  
ON L.ID_Editora = E.ID_Editora  
WHERE E.ID_Editora = 2  
ORDER BY 'Preço Ajustado' DESC;
```

Operador UNION em SQL Server

	Livro	Preço Normal	Preço Ajustado	Categoria
▶	Linux Command Line and Shell Scripting	68.35	61.5150	Tecnologia
	Using Samba	61.45	55.3050	Tecnologia
	Fedora and Red Hat Linux	62.24	56.0160	Tecnologia
	Windows Server 2012 Inside Out	66.80	60.1200	Tecnologia
	Practical Electronics for Inventors	67.80	61.0200	Tecnologia

Operador UNION em SQL Server

Exemplo 02

- Retornar nomes de livros e preços dos livros.
- Caso o preço do livro seja igual ou superior a R\$ 60,00, mostrar a mensagem “Livro Caro” em uma coluna à direita no resultado da consulta.
- Caso contrário, mostrar a mensagem “Preço Razoável”
- Ordenar por preço, do mais barato para o mais caro.

Operador UNION em SQL Server

```
SELECT Nome_Livro Livro, Preço_Livro Preço, 'Livro Caro' Resultado  
FROM tbl_Livro
```

```
WHERE Preço_Livro >= 60
```

```
UNION
```

```
SELECT Nome_Livro Livro, Preço_Livro Preço, 'Preço Razoável'  
Resultado
```

```
FROM tbl_Livro
```

```
WHERE Preço_Livro < 60
```

```
ORDER BY Preço;
```

	Livro	Preço	Resultado
►	Microsoft Exchange Server 2010	47.20	Preço Razoável
	SSH, o Shell Seguro	58.30	Preço Razoável
	Using Samba	61.45	Livro Caro
	Fedora and Red Hat Linux	62.24	Livro Caro
	Windows Server 2012 Inside Out	66.80	Livro Caro
	Practical Electronics for Inventors	67.80	Livro Caro
	Linux Command Line and Shell Scripting	68.35	Livro Caro

SQL Server – Select com dados de tabelas diferentes

SELECT

tbl_livro.Nome_Livro,tbl_livro.ID_Livro,tbl_autores.ID_Autor,tbl_autores.
Nome_Autor,tbl_autores.Sobrenome_Autor,tbl_editoras.Nome_Editora
from tbl_livro,tbl_autores,tbl_editoras
where tbl_livro.ID_Livro =5 and tbl_livro.ID_Autor = tbl_autores.ID_Autor
and tbl_livro.ID_Editora = tbl_editoras.ID_Editora;

	Nome_Livro	ID_Livro	ID_Autor	Nome_Autor	Sobrenome_Autor	Nome_Editora
▶	Windows Server 2012 Inside Out	5	4	William	Stanek	Microsoft Press

SQL Server – VIEWS – Criando Tabelas Virtuais (Visões)

Uma **View** (Exibição / Visão) é uma tabela virtual (estrutura de dados) baseada no conjunto de resultados de uma consulta SQL, criada a partir de um conjunto de tabelas (ou outras views) presentes no banco, que servem como *tabelas-base*.

SQL Server – VIEWS – Criando Tabelas Virtuais (Visões)

Contém linhas e colunas como uma tabela real, e pode receber comandos como declarações JOIN, WHERE e funções como uma tabela normal. A view fica armazenada no banco de dados.

Mostra sempre resultados de dados atualizados, pois o motor do banco de dados recria os dados toda vez que um usuário consulta a visão.

SQL Server – VIEWS – Criando Tabelas Virtuais (Visões)

Aplicações das Views

- ◎ Para simplificar o acesso a dados que estão armazenados em múltiplas tabelas relacionadas
- ◎ Implementar segurança nos dados de uma tabela, por exemplo criando uma visão que limite os dados que podem ser acessados, por meio de uma cláusula WHERE
- ◎ Prover isolamento de uma aplicação da estrutura específica de tabelas do banco acessado.

SQL Server – VIEWS – Criando Tabelas Virtuais (Visões)

Sintaxe:

```
CREATE VIEW Nome_visão  
AS SELECT colunas  
FROM tabela  
WHERE condições;
```

SQL Server – VIEWS – Criando Tabelas Virtuais (Visões)

Vamos criar uma view de nome **vw_LivrosAutores** que retorne ao ser consultada os nomes dos livros e seus respectivos autores, eliminando a necessidade de realizar INNER JOIN toda vez que precisarmos desses dados:

SQL Server – VIEWS – Criando Tabelas Virtuais (Visões)

```
CREATE VIEW vw_LivrosAutores  
AS SELECT tbl_Livro.Nome_Livro AS Livro,  
tbl_autores.Nome_Autor AS Autor  
FROM tbl_Livro  
INNER JOIN tbl_autores  
ON tbl_Livro.ID_Autor = tbl_autores.ID_Autor;
```

SQL Server – VIEWS – Criando Tabelas Virtuais (Visões)

Como usar a View criada:

```
SELECT Livro, Autor  
FROM vw_LivrosAutores  
ORDER BY Autor;
```

Livro	Autor
SSH, o Shell Seguro	Daniel
Using Samba	Gerald
Fedora and Red Hat Linux	Mark
Linux Command Line and Shell Scripting	Richard
Practical Electronics for Inventors	Simon
Windows Server 2012 Inside Out	William
Microsoft Exchange Server 2010	William

SQL Server – VIEWS – Criando Tabelas Virtuais (Visões)

Como alterar uma View:

```
ALTER VIEW vw_LivrosAutores AS  
SELECT tbl_Livro.Nome_Livro AS Livro,  
tbl_autores.Nome_Autor AS Autor, Preco_Livro AS  
Valor  
FROM tbl_Livro  
INNER JOIN tbl_autores  
ON tbl_Livro.ID_Autor = tbl_autores.ID_Autor;
```

SQL Server – VIEWS – Criando Tabelas Virtuais (Visões)

Podemos testá-la com uma consulta:

```
SELECT * FROM vw_LivrosAutores;
```

	Livro	Autor	Valor
▶	Linux Command Line and Shell Scripting	Richard	68.35
	SSH, o Shell Seguro	Daniel	58.30
	Using Samba	Gerald	61.45
	Fedora and Red Hat Linux	Mark	62.24
	Windows Server 2012 Inside Out	William	66.80
	Microsoft Exchange Server 2010	William	47.20
	Practical Electronics for Inventors	Simon	67.80

SQL Server – VIEWS – Criando Tabelas Virtuais (Visões)

Como excluir views:

```
DROP VIEW vw_LivrosAutores;
```


Concatenação de Strings com as funções CONCAT e COALESCE em SQL Server

Para estudarmos estas funções, vamos criar uma tabela em um banco de dados de teste. Use o código a seguir para isso:

```
CREATE TABLE teste_nulos (  
id smallint PRIMARY KEY IDENTITY ,  
item varchar(20),  
quantidade smallint NULL);
```

Concatenação de Strings com as funções CONCAT, COALESCE em SQL Server

Inserir dados para testes:

```
INSERT INTO teste_nulos (item, quantidade)  
VALUES ('Pendrive', 5);
```

```
INSERT INTO teste_nulos (item, quantidade)  
VALUES ( 'Monitor', 7);
```

```
INSERT INTO teste_nulos (item, quantidade)  
VALUES ('Teclado', 8);
```

Função CONCAT

Sintaxe:

CONCAT(string ou nome_coluna, <string |
nome_coluna)

Função CONCAT - Exemplos

```
SELECT CONCAT('Rodrigo ', 'Vilela') AS 'Meu Nome';
```

```
SELECT CONCAT(Nome_autor, ' ', Sobrenome_autor) AS 'Nome  
Completo'  
FROM tbl_autores;
```

```
SELECT CONCAT('Eu gosto do livro ', Nome_Livro)  
FROM tbl_Livro WHERE ID_livro = 3;
```

Concatenação com NULL

```
SELECT CONCAT('A quantidade adquirida é ', '',  
quantidade)  
FROM teste_nulos  
WHERE item = 'Teclado';
```

Função COALESCE

```
SELECT CONCAT('A quantidade adquirida é ', '',  
COALESCE(NULL, quantidade, NULL, 0))  
FROM teste_nulos  
WHERE item = 'Teclado';
```

Essa função retornará o primeiro valor não-nulo encontrado em seus argumentos.