**JOINS e INNER JOIN – Selecionar dados de duas ou mais tabelas – SQL Server – Aula 22**

**JOINS e INNER JOIN – Selecionar dados de duas ou mais tabelas no SQL Server**

Empregamos as cláusulas **JOIN** (junções) para combinar dados armazenados em duas ou mais tabelas em um banco de dados, baseados em um relacionamento entre colunas destas tabelas.

Existem duas categorias principais de joins:

**INNER JOIN:** Mostra linhas (registros) quando houver pelo menos uma correspondência em ambas as tabelas.

**OUTER JOIN:** Mostra linhas (registros) mesmo quando não houver ao menos uma correspondência em uma das tabelas (ou ambas). A cláusula OUTER JOIN posusi os subtipos **RIGHT JOIN**, **LEFT JOIN** e **FULL JOIN**.

Nesta aula vamos estudar a cláusula **INNER JOIN**.

Cláusula **INNER JOIN**

Como vimos, a cláusula INNER JOIN (ou ainda simplesmente JOIN) nos permite obter registros com dados provenientes de duas ou mais tabelas que sejam relacionadas no banco de dados.  
A sintaxe básica de um INNER JOIN em uma consulta é:

**SELECT colunas**

**FROM tabela1**

**INNER JOIN tabela2**

**ON tabela1.coluna = tabela2.coluna;**

Onde t**abela1.coluna**é o nome da primeira tabela concatenado com um ponto 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.

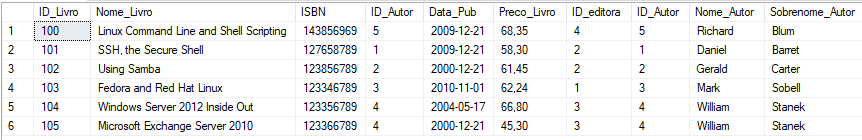
**Exemplos**

Vamos a um exemplo. Consultaremos as tabelas de livros e de autores (tbl\_livros e tbl\_autores) e vamos retornar os registros relativos aos livros e aos autores simultaneamente:

**SELECT \* FROM tbl\_livros**

**INNER JOIN tbl\_autores**

**ON tbl\_livros.ID\_Autor = tbl\_autores.ID\_Autor;**



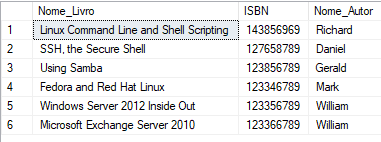
Vamos a outro exemplo. Consultaremos as tabelas de livros e autores, porém retornando apenas os nomes dos livros, seus ISBNs e os nomes dos respectivos autores desses livros. Note que as duas primeiras informações estão na tabela de livros, e o nome do autor está armazenado somente na tabela de autores. O que as conecta é o relacionamento existente entre as colunas ID\_Autor existentes em ambas as tabelas – chave primária na tabela de autores e chave estrangeira na tabela de livros:

**SELECT tbl\_livros.Nome\_Livro, tbl\_livros.ISBN, tbl\_autores.Nome\_Autor**

**FROM tbl\_livros**

**INNER JOIN tbl\_autores**

**ON tbl\_livros.ID\_Autor = tbl\_autores.ID\_Autor;**



Mais um exemplo, agora usando filtros com a cláusula **WHERE**. Vamos retornar os nomes dos livros e nomes das editoras, mas somente das editoras cujo nome se inicie com a letra M. Perceba o emprego de aliases (apelidos) nestas declarações, para simplificar o código:

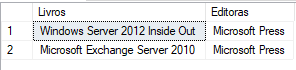
**SELECT L.Nome\_Livro AS Livros, E.Nome\_editora AS Editoras**

**FROM tbl\_livros AS L**

**INNER JOIN tbl\_editoras AS E**

**ON L.ID\_editora = E.ID\_editora**

**WHERE E.Nome\_Editora LIKE 'M%';**



Vejamos um quarto exemplo para finalizar. Agora faremos um **INNER JOIN** usando as três tabelas do banco de dados relacionadas. Desejamos os nomes e preços dos livros, nomes de seus autores e editoras, porém somente das editoras cujo nome se inicie com a letra O, sendo tudo isso ordenado em ordem crescente de preço dos livros (do mais barato para o mais caro):

**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\_livros 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 ASC;**



Neste exemplo fazemos o join entre a tabela de livros e a de autores, e logo na sequência o join entre a tabela de livros e a de editoras, pois os relacionamentos ocorrem entre essas tabelas. No final também ordenamos os resultados por ordem crescente dos preços dos livros.