

# Curso Básico de MySQL

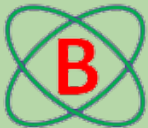
Junção de Tabelas com INNER JOIN



# JOINS

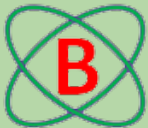
Cláusulas JOIN são usadas para combinar dados provenientes de duas ou mais tabelas em um único conjunto de resultados, baseado em condições de join especificadas.

Há duas sintaxes possíveis para codificar joins: sintaxe explícita (SQL-92; recomendada) e sintaxe implícita (condição de join na cláusula WHERE).



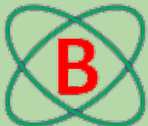
# JOINS - Categorias

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



# Cláusula ON

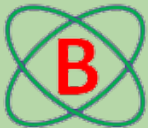
- A cláusula **ON** determina a condição de join, que indica como as tabelas devem ser comparadas.
- No geral, a comparação ocorre por meio de um relacionamento entre chave primária na primeira tabela e chave estrangeira na segunda tabela.



# Condição de Join

Uma condição de join nomeia uma coluna em cada tabela envolvida no join e indica como as colunas devem ser comparadas.

No geral, usamos o operador = para obter linhas com colunas correspondentes. É comum usar o relacionamento de PK de uma tabela com FK de outra tabela.



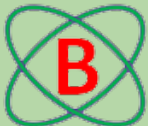
# Nomes de Coluna Qualificados

Um nome de coluna qualificado possui o nome da coluna precedido pelo nome da tabela à qual pertence, separados por um ponto.

Exemplo:

**tbl\_Livros.IdEditora**

Indica a coluna **IdEditora** especificamente da tabela **tbl\_Livros**.

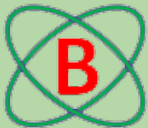


# Nomes de Coluna Qualificados

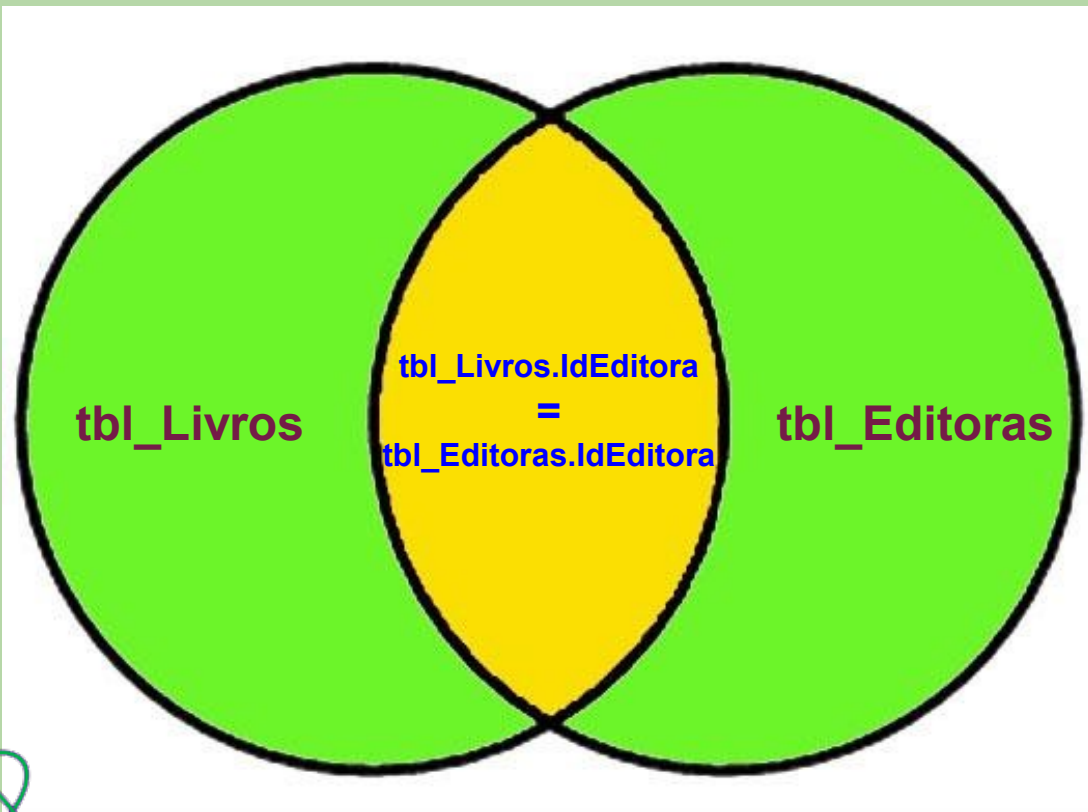
Usamos os nomes de colunas qualificados em JOINS para identificar exatamente a qual tabela cada campo envolvido pertence.

Isso evita erro de ambiguidade caso uma coluna tenha o mesmo nome em duas tabelas diferentes.

**ON tbl\_Livros.IdEditora = tbl\_Editoras.IdEditora**



# INNER JOIN





# INNER JOIN

## Sintaxe

**SELECT** colunas

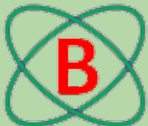
**FROM** tabela1

**[INNER] JOIN** tabela2

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

**[INNER] JOIN** tabelaN

**ON** tabela1.coluna=tabelaN.coluna;



# INNER JOIN - Exemplos

```
SELECT * FROM tbl_Livros
```

```
INNER JOIN tbl_Editoras
```

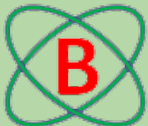
```
ON tbl_Livros.IdEditora = tbl_Editoras.IdEditora;
```

```
SELECT tbl_Livros.NomeLivro, tbl_Livros.ISBN13,
```

```
tbl_Assuntos.Assunto
```

```
FROM tbl_Livros JOIN tbl_Assuntos
```

```
ON tbl_Livros.IdAssunto = tbl_Assuntos.IdAssunto;
```



# INNER JOIN – Mais exemplos

Usando Aliases e cláusulas WHERE e LIKE:

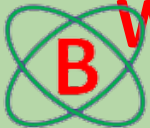
```
SELECT L.NomeLivro AS Livros, E.NomeEditora AS  
Editoras
```

```
FROM tbl_Livros AS L
```

```
JOIN tbl_Editoras AS E
```

```
ON L.IdEditora = E.IdEditora
```

```
WHERE E.NomeEditora LIKE 'M%';
```



# INNER JOIN - com três tabelas

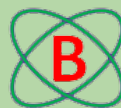
```
SELECT L.NomeLivro Livro, C.Assunto Assunto,  
       E.NomeEditora Editora  
FROM tbl_Livros L  
      JOIN tbl_Assuntos C  
        ON L.IdAssunto = C.IdAssunto  
      JOIN tbl_Editoras E  
        ON L.IdEditora = E.IdEditora;
```



# INNER JOIN - outro com três tabelas

```
SELECT L.NomeLivro Livro,  
       CONCAT(A.NomeAutor, ' ', A.SobrenomeAutor) Autor,  
       L.PrecoLivro 'Preço do Livro'  
FROM tbl_livrosautores LA  
     JOIN tbl_livros L ON L.IdLivro = LA.IdLivro  
     JOIN tbl_Autores A ON A.IdAutor = LA.IdAutor;
```





# Próximo: OUTER JOINS



# Joins com USING

Normalmente, as colunas comparadas em uma cláusula ON possuem o mesmo nome.

Quando essa comparação ocorre por meio de um ***equijoin*** (sinal de =), é possível empregar a palavra **USING** para simplificar a consulta.



# Joins com USING

Sintaxe:

**SELECT colunas**

**FROM tabela1**

**JOIN tabela2 USING (coluna\_em\_comum)**

**JOIN tabelaN USING (coluna\_em\_comum);**





# Joins com USING

```
SELECT L.NomeLivro AS Livro,  
       CONCAT(A.NomeAutor, ' ', A.SobrenomeAutor) AS  
       Autor,  
       L.PrecoLivro As 'Preço do Livro'  
FROM tbl_livrosautores LA  
     JOIN tbl_Livros L USING(IdLivro)  
     JOIN tbl_Autores A USING(IdAutor)  
WHERE L.PrecoLivro BETWEEN 160.00 AND 200.00  
ORDER BY Livro;
```

# NATURAL Joins

Em uma natural join, não especificamos a coluna usada para conectar as tabelas. O BD automaticamente as une baseado nas colunas das duas tabelas que possuem o mesmo nome.

Este tipo de join só funciona corretamente se o banco for projetado de forma específica.

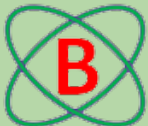
Evite o uso de joins naturais, a não ser em tabelas simples relacionadas com um único campo em comum.



# NATURAL Joins

Sintaxe:

```
SELECT colunas  
FROM tabela1  
    NATURAL JOIN tabela2  
    [NATURAL JOIN tabelaN]...;
```



# Joins com NATURAL

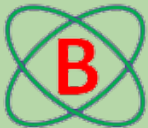
```
SELECT L.NomeLivro AS Livro,  
       CONCAT(A.NomeAutor, ' ', A.SobrenomeAutor) AS  
       Autor,  
       L.PrecoLivro As 'Preço do Livro'  
FROM tbl_livrosautores LA  
     NATURAL JOIN tbl_Livros L  
     NATURAL JOIN tbl_Autores A  
WHERE L.PrecoLivro BETWEEN 160.00 AND 200.00  
ORDER BY Livro;
```

# INNER JOIN Implícito

Forma antiga de se realizar joins. Importante conhecer para dar manutenção em bancos legados.

Nesta forma de join, as condições de join são codificadas na cláusula WHERE.

Recomenda-se NÃO usar joins implícitas em queries.



# INNER JOIN Implícito

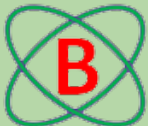
Sintaxe:

**SELECT colunas**

**FROM tabela1, tabela2 [, tabelaN]...**

**WHERE tabela1.coluna operador tabela2.coluna**

**[AND tabela2.coluna operador tabela3.coluna]...**



# INNER JOIN Implícito

```
SELECT L.NomeLivro AS Livro,  
       CONCAT(A.NomeAutor, ' ', A.SobrenomeAutor) AS  
       Autor,  
       L.PrecoLivro As 'Preço do Livro'  
FROM tbl_livrosautores LA, tbl_Livros L, tbl_Autores A  
WHERE L.IdLivro = LA.IdLivro  
       AND LA.IdAutor = A.IdAutor  
       AND L.PrecoLivro BETWEEN 160.00 AND 200.00  
ORDER BY Livro;
```