



*Serviço Nacional de Aprendizagem Industrial*

**PELO FUTURO DO TRABALHO**

# Modelagem de banco de dados relacional: modelagem lógica e física.

Rafael C. Ventura

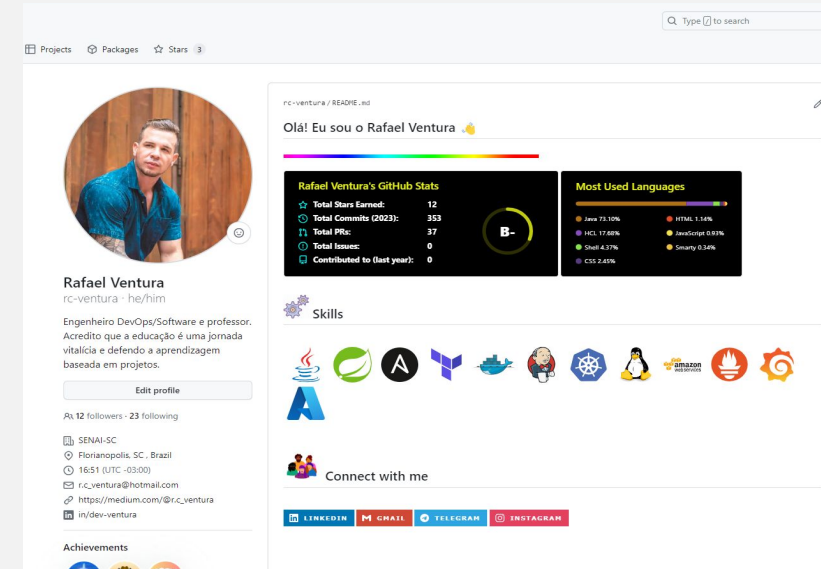
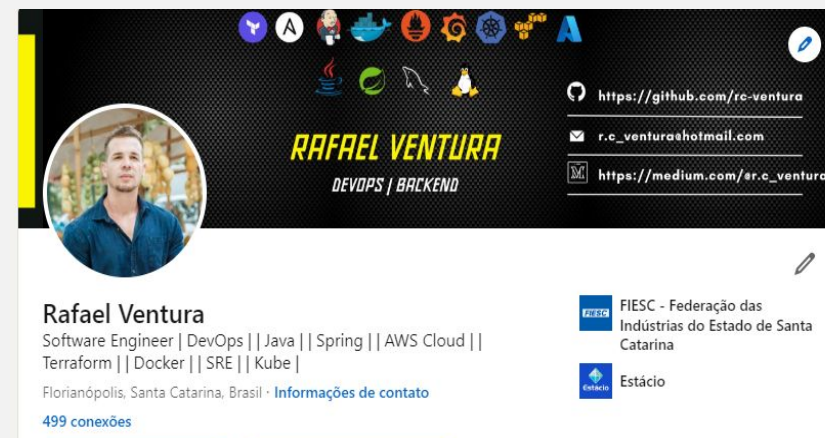
# Quem sou eu?

- Rafael C. Ventura
- Técnico em Segurança de Dados pela ANHANGUERA.
- Estudo Análise e Desenvolvimento de Sistemas na Estácio.
- Mestre em Educação pela PUC-RJ
- Leciono há mais de 10 anos.
- Experiência Cloud, IaC, DevOps
- email: rafael.ventura@edu.sc.senai.br




# Redes Sociais

- ❑ <https://www.linkedin.com/in/dev-ventura/>
- ❑ <https://github.com/rc-ventura>



# Entra 21

- ☐ Lógica com Java Script ( 4 encontros) 
- ☐ Banco de dados Relacionais (10 encontros)
- ☐ Metodologia Ágeis (1 encontro)
- ☐ Github (3 encontros)
- ☐ Programação Orientada à Objetos JS ( 10 encontros)
- ☐ React (15 encontros)
- ☐ Desenvolvimento do Trabalho de Conclusão do Curso ( 6 encontros)

# O que iremos aprender?

- ❑ Comandos SQL
- ❑ Utilizar o SELECT, FROM
- ❑ Manipular os operadores com WHERE
- ❑ Utilizar comandos de ordenação como GROUP BY e ORDER BY
- ❑ Fazer consultas usando o comando HAVING

# Nossas referências

□ **RelaX** (<https://dbis-uibk.github.io/relax/landing>)

□ **SCALER** (<https://www.scaler.com/topics/sql/sql-query-execution-order/>)

# Projeto Desafio: Banco de Dados Clube do Livro

- Iremos traduzir alguns comandos já aprendidos para linguagem SQL.



Inspector Navegação Configuração

Versão

Versão do diagrama	3.2.0
Nome	projeto_aula_clube_do_l
Arquivo	projeto_aula_clube_do_l
Autor(es)	
Observações	
Diagrama	Conceitual

Dimensões, cor e etc.

Largura	4096
Altura	4096
Zoom	100.0%

Espaço para alinhamento

Espaço horizontal	60
Espaço vertical	50

Fonte

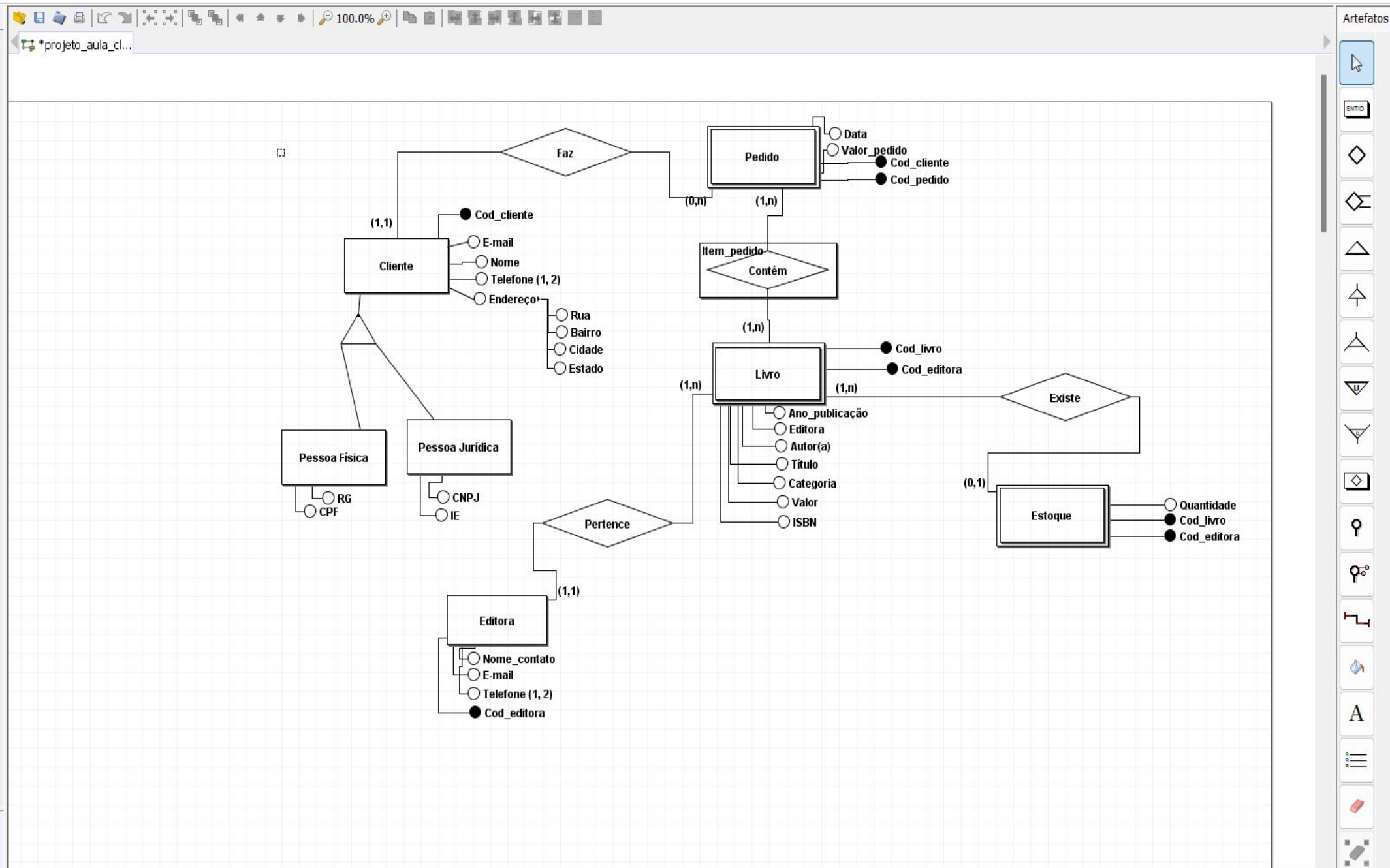
Nome fonte	Arial
Tamanho da fonte	12
Estilo da fonte	Estilo negrito
Editar fonte	Arial Negrito

Converter

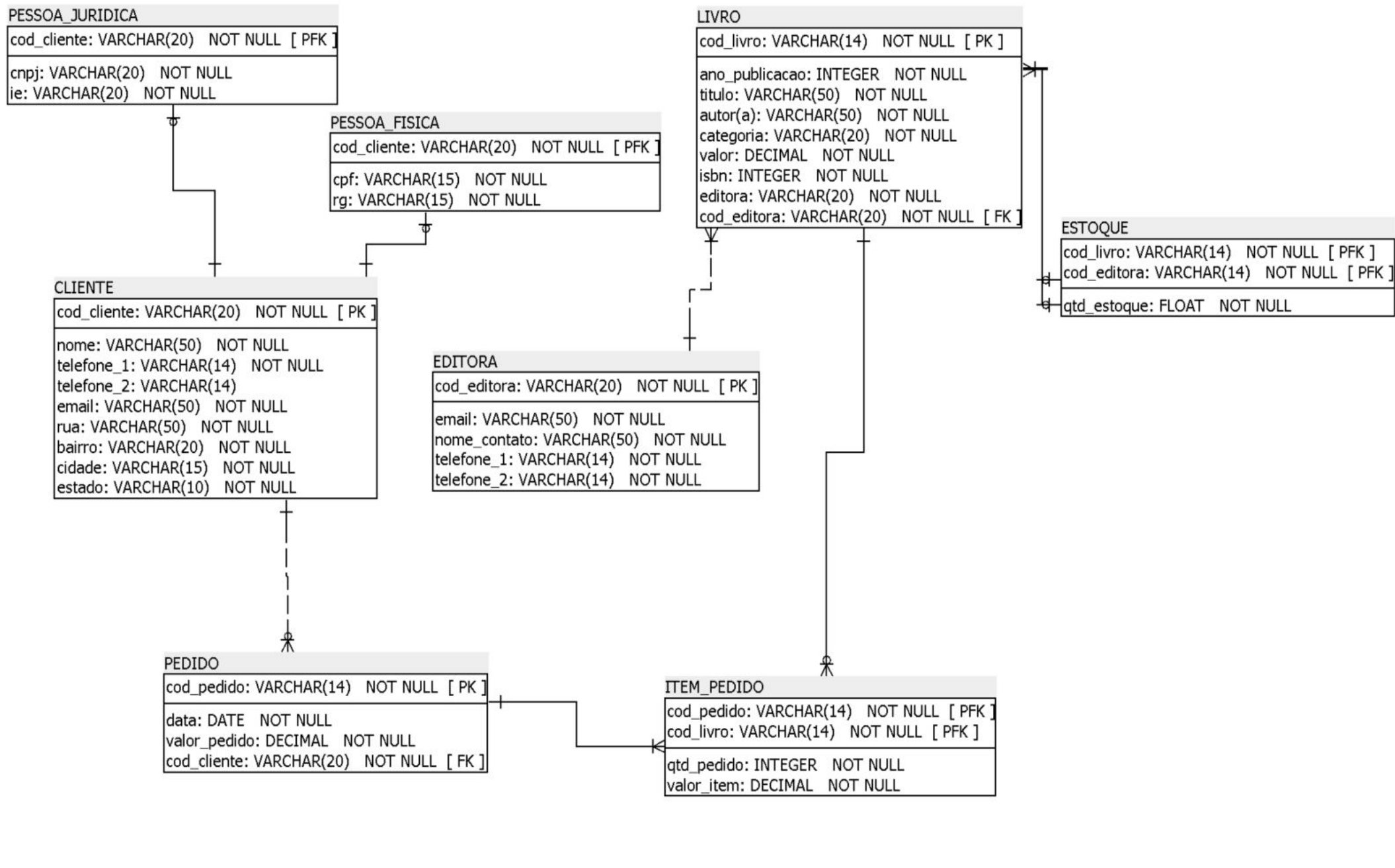
Converter	...
-----------	-----

Editar atributos

Editar atributos	...
------------------	-----







SELECT

FROM

WHERE

# Comandos SQL

- **SELECT** = Filtra colunas ( $\pi$ )
- **FROM** = Seleciona as tabelas
- **WHERE** = Filtra linhas sob alguma condição ( $\sigma$ )

# Comandos SQL

□ **GROUP BY** = Agrupa linhas de uma tabela com base em valores em uma ou mais colunas;

- Ex: (Quero agrupar por nome do cliente).
- Isso auxilia para processamento com outras cláusulas.
- Ajuda a resumir dados.
- Associado às operações SUM, AVG, COUNT, MAX, and MIN

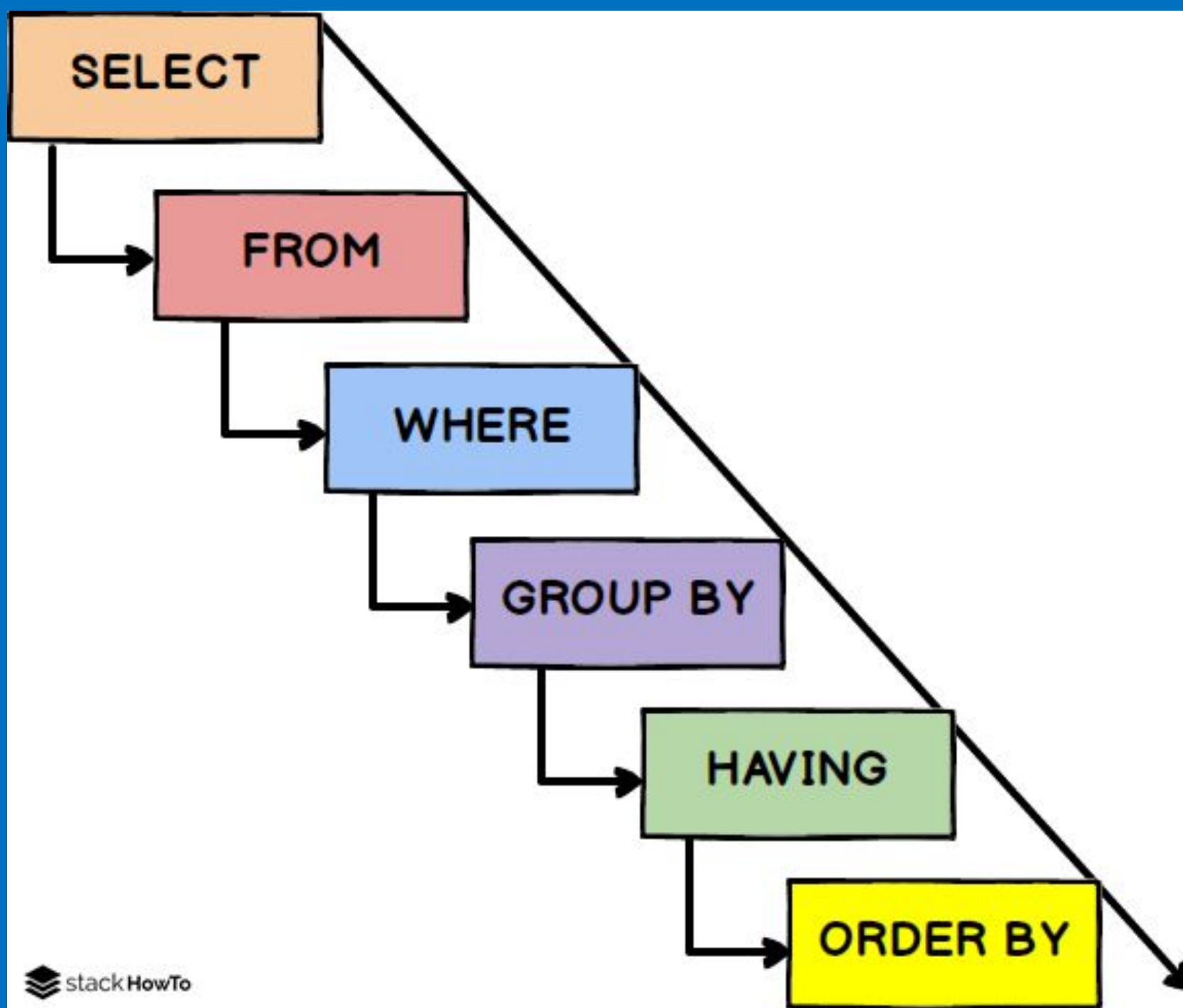
□ **HAVING** = filtra grupos para uma condição.

- É usado com o **GROUP BY**
- (Ex: Após agrupar os produtos por categoria (**GROUP BY**), quero filtrar produtos com quantidade maior que 3)

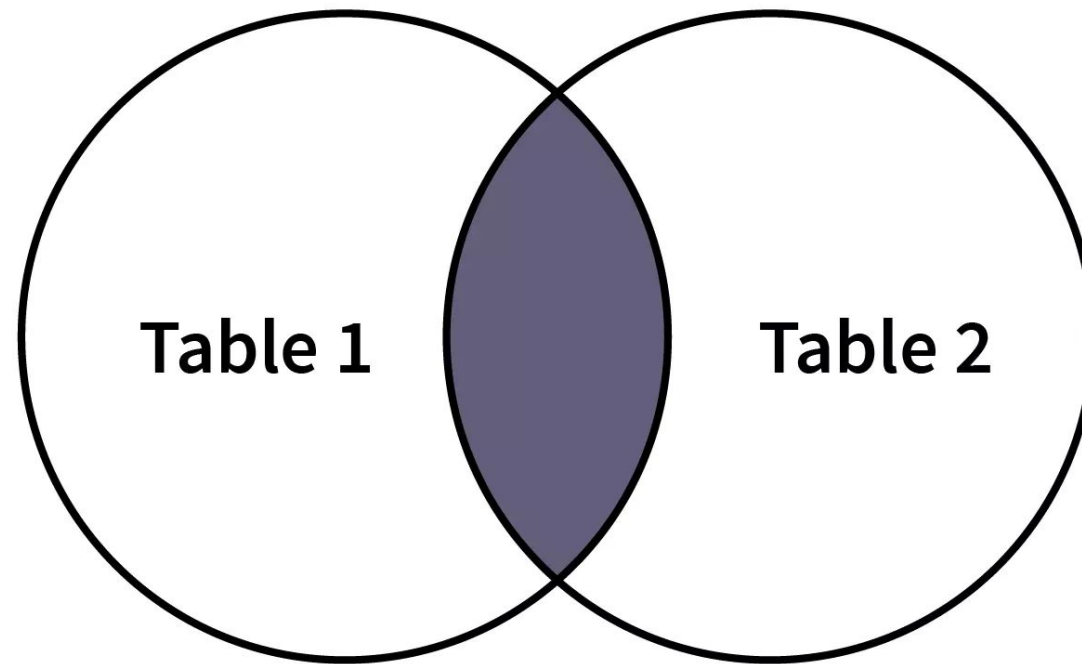
# Comandos SQL

- **ORDER BY** = Ordena por base a uma ou mais colunas (ASC/DESC)
- **LIMIT** = limita a quantidade de linhas para o resultado









**INNER JOIN**

**Table\_Name : Customers**

CustID	Name	Phone_Number
1	Raj Mehta	98540XXXXX
2	Sanjay Mishra	88888XXXXX
3	Aditi Gupta	67809XXXXX
4	Manish Chopra	12345XXXXX

**Table\_Name : Shopping\_Details**

ItemID	CustID	Item_Name	Quantity
1	2	Chips	2
2	3	Chocolate	5
3	5	Dress	8

 **Primary key**  
 **Foreign key**

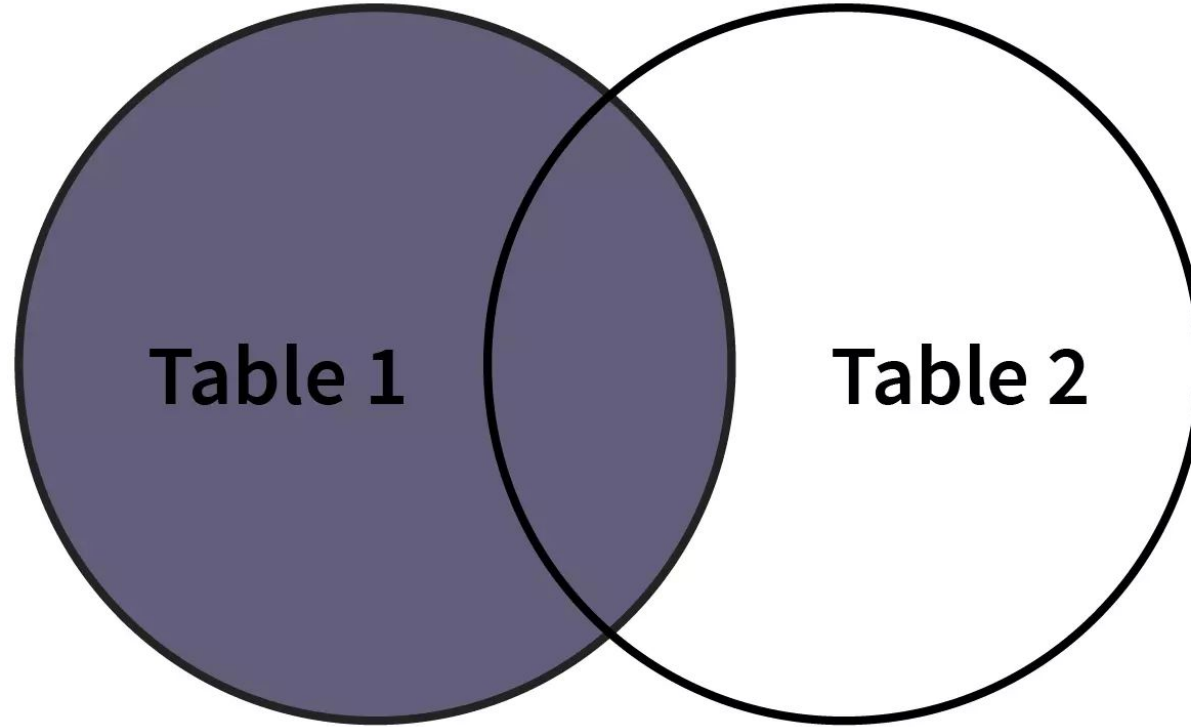
**Temporary Table**

Customers CustID	Name	Phone_Number	Item_ID	Shopping_d etails.CustID	Item_Name	Quantity
1	Raj Mehta	98540XXXXX	NULL	NULL	NULL	NULL
2	Sanjay Mishra	88888XXXXX	1	2	Chips	2
3	Aditi Gupta	67809XXXXX	2	3	Chocolate	5
4	Manish Chopra	12345XXXXX	NULL	NULL	NULL	NULL
NULL	NULL	NULL	3	5	Dress	8

**Resultant Table**

Name	Item_Name	Quantity
Sanjay Mishra	Chips	2
Aditi Gupta	Chocolate	5

```
SELECT Customers.Name,
Shopping_Details.Item_Name,
Shopping_Details.Quantity
FROM Customers INNER JOIN
Shopping_Details
WHERE
Customers.ID==Shopping_Details.ID;
```



**LEFT JOIN**

**Table\_Name : Customers Left**

CustID	Name	Phone_Number
1	Raj Mehta	98540XXXXX
2	Sanjay Mishra	88888XXXXX
3	Aditi Gupta	67809XXXXX
4	Manish Chopra	12345XXXXX

**Table\_Name : Shopping\_Details Right**

ItemID	CustID	Item_Name	Quantity
1	2	Chips	2
2	3	Chocolate	5
3	5	Dress	8



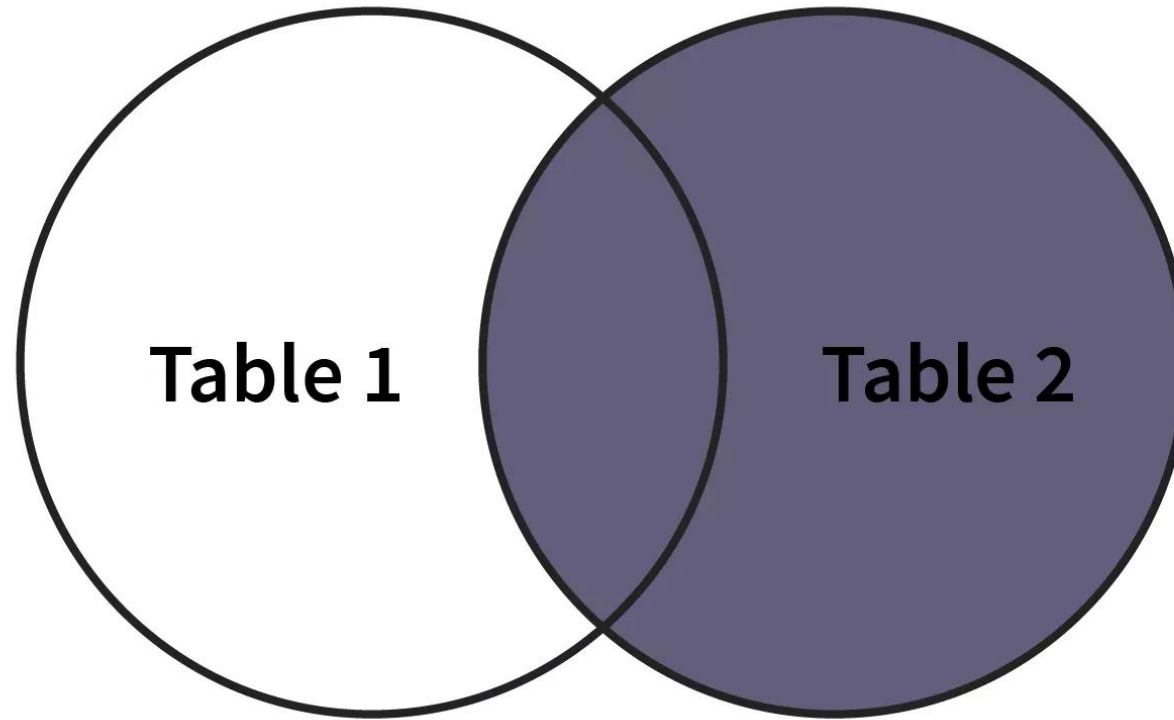
**Temporary Table**

Customers CustID	Name	Phone_Number	Item_ID	Shopping_d etails.CustID	Item_Name	Quantity
1	Raj Mehta	98540XXXXX	NULL	NULL	NULL	NULL
2	Sanjay Mishra	88888XXXXX	1	2	Chips	2
3	Aditi Gupta	67809XXXXX	2	3	Chocolate	5
4	Manish Chopra	12345XXXXX	NULL	NULL	NULL	NULL
NULL	NULL	NULL	3	5	Dress	8

**Resultant Table**

Name	Item_Details
Sanjay Mishra	Chips
Aditi Gupta	Chocolate
Raj Mehta	NULL
Manish Chopra	NULL

```
SELECT Customers. Name,
Shopping_Details.Item_Name
FROM Customers LEFT OUTER JOIN
Shopping_Details;
ON Customers.ID =
Shopping_Details.ID;
```



**RIGHT JOIN**

**Table\_Name : Customers** **Left**

CustID	Name	Phone_Number
1	Raj Mehta	98540XXXXX
2	Sanjay Mishra	88888XXXXX
3	Aditi Gupta	67809XXXXX
4	Manish Chopra	12345XXXXX

**Table\_Name : Shopping\_Details** **Right**

ItemID	CustID	Item_Name	Quantity
1	2	Chips	2
2	3	Chocolate	5
3	5	Dress	8



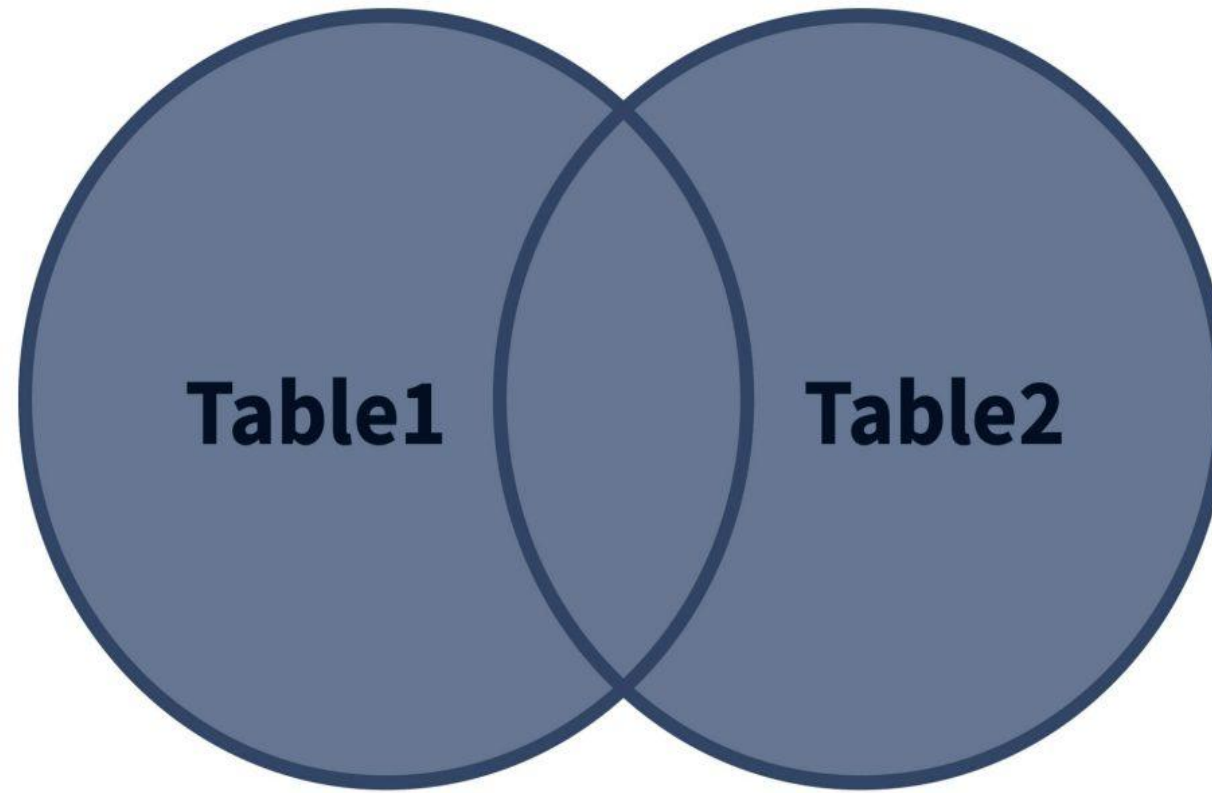
**Temporary Table**

Customers CustID	Name	Phone_Number	Item_ID	Shopping_d etails.CustID	Item_Name	Quantity
1	Raj Mehta	98540XXXXX	NULL	NULL	NULL	NULL
2	Sanjay Mishra	88888XXXXX	1	2	Chips	2
3	Aditi Gupta	67809XXXXX	2	3	Chocolate	5
4	Manish Chopra	12345XXXXX	NULL	NULL	NULL	NULL
NULL	NULL	NULL	3	5	Dress	8

**Resultant Table**

Name	Item_Details
Sanjay Mishra	Chips
Aditi Gupta	Chocolate
NULL	Dress

```
SELECT Customers.Name,
Shopping_Details.Item_Name
FROM Customers RIGHT OUTER JOIN
Shopping_Details;
ON Customers.ID =
Shopping_Details.ID;
```



**FULL JOIN**



**Table Name: Customers**

CustID	Name	Phone_Number
1	Raj Mehta	98540XXXXX
2	Sanjay Mishra	88888XXXXX
3	Aditi Gupta	67809XXXXX
4	Manish Chopra	12345XXXXX

**Table Name: Shopping\_Details**

ItemID	CustID	Item_Name	Quantity
1	2	Chips	2
2	3	Chocolate	5
3	5	Dress	8

Primary Key  
 Foreign Key

**Temporary table**

Customers CustID	Name	Phone_Number	ItemID	Shopping_D etails.CustID	Item_Name	Quantity
1	Raj Mehta	98540XXXXX	NULL	NULL	NULL	NULL
2	Sanjay Mishra	88888XXXXX	1	2	Chips	2
3	Aditi Gupta	67809XXXXX	2	3	Chocolate	5
4	Manish Chopra	12345XXXXX	NULL	NULL	NULL	NULL
NULL	NULL	NULL	3	5	Dress	8

**Resultant table**

Name	Item_Details
Sanjay Mishra	Chips
Aditi Gupta	Chocolate
Raj Mehta	NULL
Manish Chopra	NULL
NULL	Dress

```

SELECT Customers.Name,
Shopping_Details.Item_Name
FROM Customers FULL OUTER
JOIN Shopping_Details
WHERE Customer.ID =
Shopping_Details.ID;
    
```



# Comandos SQL

- **INNER JOIN** = Combina linhas de duas tabelas com base em uma condição de correspondência. Ele retorna apenas as linhas que têm correspondências em ambas as tabelas.
- **LEFT JOIN** = combina todas as linhas da tabela da esquerda (tabela principal) com as linhas correspondentes da tabela da direita (tabela secundária). Se não houver correspondência na tabela da direita, ele retorna valores nulos para as colunas da tabela da direita.

# Comandos SQL

- ❑ **RIGHT JOIN =** É o oposto da LEFT JOIN. Ela combina todas as linhas da tabela da direita com as linhas correspondentes da tabela da esquerda. Se não houver correspondência na tabela da esquerda, ele retorna valores nulos para as colunas da tabela da esquerda.
- ❑ **FULL JOIN =** combina todas as linhas de ambas as tabelas, retornando todas as linhas da tabela da esquerda e da tabela da direita. Se não houver correspondência em uma tabela, ele retorna valores nulos para as colunas da outra tabela.

# WHERE

OPERADOR	DESCRIÇÃO
=	Igual
>=	Maior e Igual que
<=	Menor e igual que
>	Maior que
<	Menor que
<>	Não igual
BETWEEN	Entre
Like	Procura um padrão semelhante

# Comandos SQL

- **ALIAS** = Dar um apelido tanto para colunas quanto para tabelas.
  - Simplifica a manipulação das consultas
  - Ex: A tabela '**Cliente**' posso chamar de '**c**'.



*Serviço Nacional de Aprendizagem Industrial*

**PELO FUTURO DO TRABALHO**

**0800 048 1212**     **sc.senai.br**

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