



# Auto estudo...

Até 10h





# DAILY

## A FAZER



## DESENVOLVIMENTO - 5



## TESTE - 3



## CONCLUÍDO



Até 10h15

## INTRODUÇÃO À COMPUTAÇÃO

# Banco de Dados III JOINS





# Agenda do dia



Explicação Teórica



Mão na massa





# QUEM SOU EU...

Dr. CRISTIANO  
BENITES









*Deficiente auditivo \*Grau profundo*

**\*CID H91.8**

*Especialista em leitura labial*



# Combinados

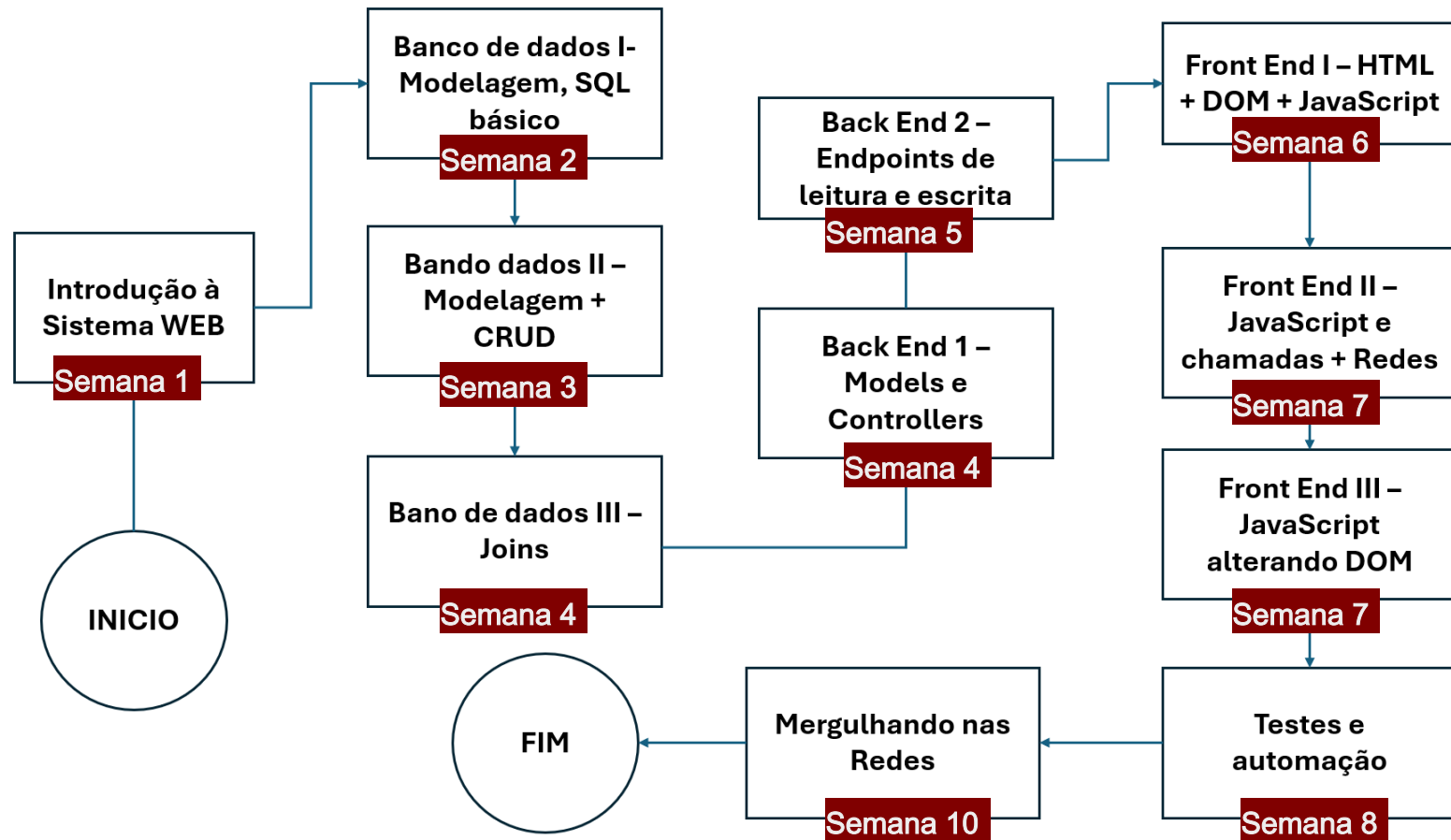
-  Você é responsável pela seu Check in na Adalove
-  Presença na aula das 10h até 12h. “Ausência acarretará em falta”
-  Atividades entregues no prazo
-  Estou sempre disponível no SLACK ou em sala de aula
-  Se planejem sempre de forma antecipada
-  Avaliem nosso encontro na Adalove. Vamos praticar feedback o tempo todo







# Diagrama Módulo






# Conteúdo do dia




 Banco de Dados III - Joins


Semana 04

 Consultas em SQL - Parte II

Semana 04

 Trabalhando com várias tabelas em bancos de dados

Semana 04

 Entendendo chave primária e estrangeira nos bancos de dados

Semana 04

 SQL for beginners: how to get started with JOINS

Semana 04

 Tutorial SQL

Semana 04



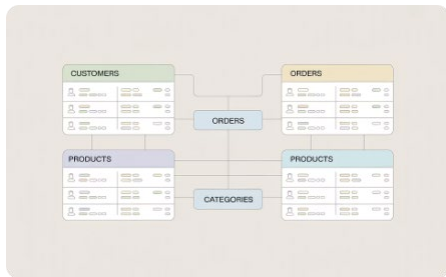


# Banco de Dados III: Trabalhando com JOINS em SQL

Consultas que conectam tabelas para extrair informações completas.

Técnicas avançadas para manipular dados relacionais.

# Relações Entre Tabelas em Bancos de Dados



## Banco Relacional

Divide dados em tabelas especializadas.

## Por que múltiplas tabelas?

Evita redundância de dados

Facilita manutenção

Melhora performance



## Tabelas Relacionadas

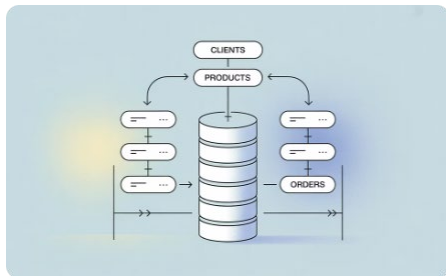
Clientes ligados a pedidos e produtos.

## Nosso exemplo

Clientes: dados pessoais

Pedidos: compras realizadas

Produtos: itens disponíveis



## Sem Redundância

Evita duplicação de informações.

# Chave Primária: Conceito e Importância



Identificador único

Cada registro precisa ser  
único



Exemplo: id\_cliente

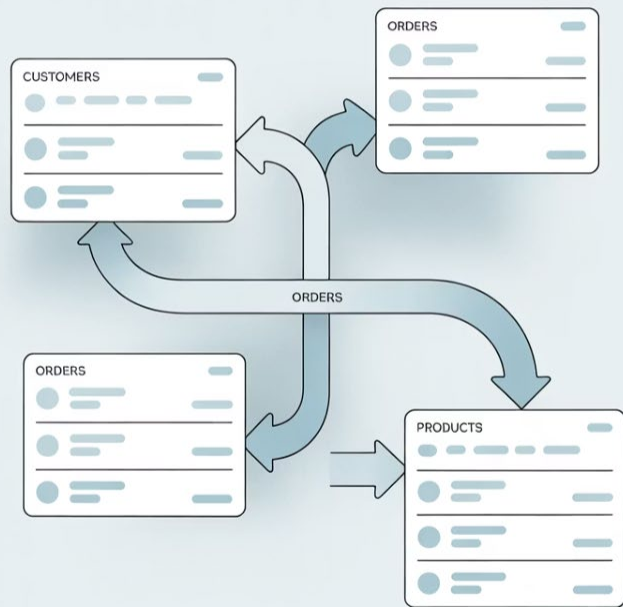
Número exclusivo para cada  
cliente



Não permite duplicatas

Garante integridade dos dados





# Chave Estrangeira: Relacionando Tabelas



## Chave Primária em Clientes

id\_cliente único identifica cada pessoa.



## Chave Estrangeira em Pedidos

id\_cliente liga pedido ao respectivo cliente.



## Integridade Referencial

Não permite orfãos no banco.

# Porque Realizar Consultas com Junções (JOINS)



Integra informações dispersas

Combina dados que fazem sentido juntos



Evita redundância

Mantém dados em seu lugar ideal



Consultas complexas

Responde perguntas complexas sobre os dados



Mantém consistência

Dados atualizados em um único lugar

Table A

Inner  
Join

Table B

## Compreendendo o INNER JOIN

1

2

3

Tabela A

Primeira fonte de dados

Correspondência

Registros que existem em ambas

Tabela B

Segunda fonte de dados

```
SELECT * FROM TabelaA INNER JOIN TabelaB ON A.chave = B.chave
```

← 🔍 Slenderid 4-0902012 ⓘ 43.103 ⚡ 11.262-1808 🗨️ 901.29.1 7209

🔒 Customerid ➕ Adicionar cliente 📄 1-200 7809. 50902 📌 1 D 014. 21991120 Customerdate ⓘ

Orderid	Order17	Order1,2

## Exemplo Prático: INNER JOIN

Nome	Email	Data Pedido	Valor
Ana Silva	ana@email.com	10/09/2023	R\$150,00
Carlos Lima	carlos@email.com	15/09/2023	R\$300,00

```
SELECT c.nome, c.email, p.data, p.valor FROM Clientes c INNER JOIN Pedidos p ON c.id = p.id_cliente
```



# LEFT JOIN (ou LEFT OUTER JOIN)



## Tabela Esquerda Completa

Todos registros da primeira tabela aparecem.

---



## NULL para Não Correspondentes

Campos vazios quando não há correspondência.

---



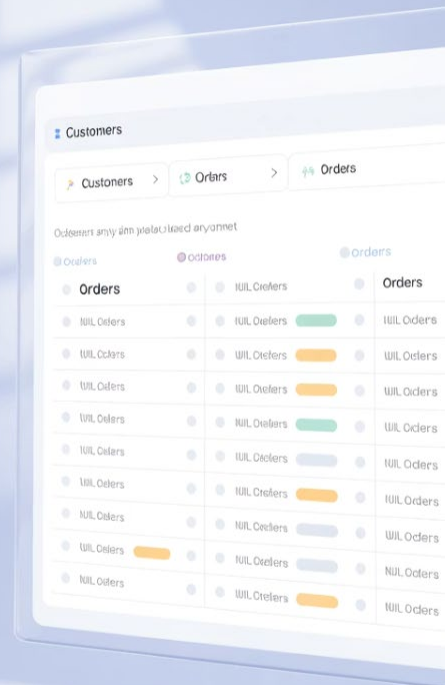
## Sintaxe

```
SELECT * FROM A LEFT JOIN B ON A.id = B.id
```

# Exemplo Prático: LEFT JOIN

Cliente	Data Pedido	Valor
Ana Silva	10/03/2023	R\$150,00
Bruno Costa	NULL	NULL
Carla Mendes	15/03/2023	R\$320,00

SQL Join  
Example



# RIGHT JOIN (ou RIGHT OUTER JOIN)



Inclui todos da tabela à direita

Mesmo sem correspondência



Oposto do LEFT JOIN

Alterna o foco da consulta



Mesmo resultado trocando tabelas

LEFT JOIN inverso = RIGHT JOIN

# FULL JOIN (ou FULL OUTER JOIN)



União Completa (Junta tudo)

Registros de ambas tabelas aparecem no resultado.



NULL para Não Correspondentes

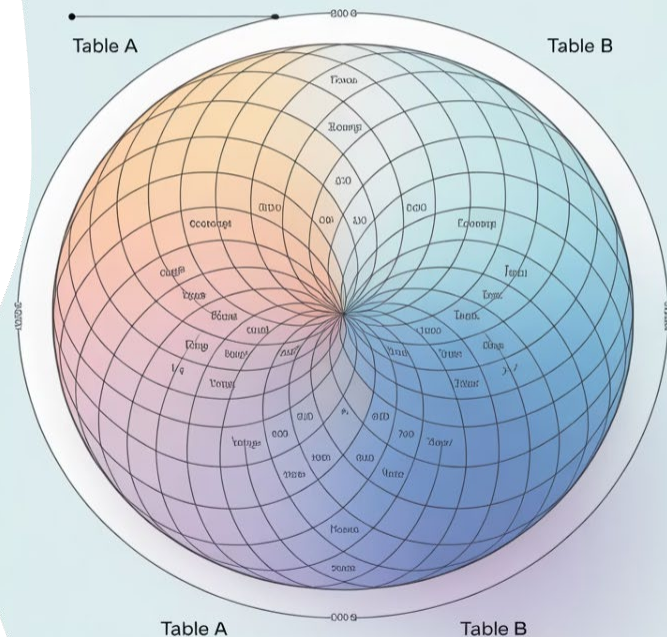
Preenche com NULL campos sem correspondência.



Análises Abrangentes (Análise completa)

Ideal para verificar inconsistências entre tabelas.

## “Full Outer Join Visualization”



# JOINS com Mais de Duas Tabelas



# Boas Práticas e Otimização em Consultas JOINS



## Crie índices

Em chaves usadas em  
JOINS



Selecione apenas  
colunas necessárias

Evite SELECT \*



Teste  
performance

Use EXPLAIN para  
analisar



Filtre antes de  
juntar

WHERE antes de JOIN



# Erros Comuns ao Trabalhar com JOINS

01

Ambiguidade de Coluna

Erro de coluna ambígua por não especificar a tabela.

02

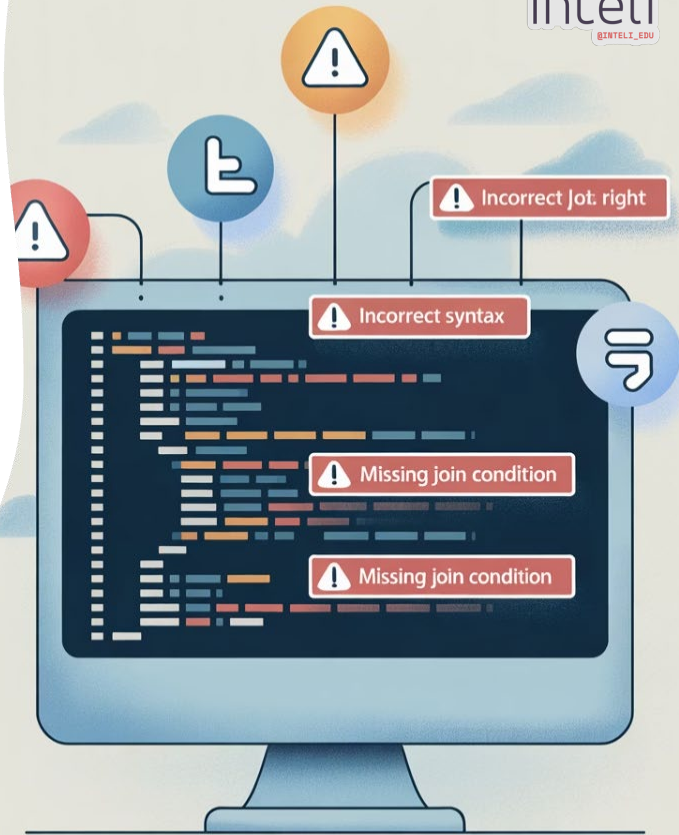
CROSS JOIN Acidental

Esquecimento da condição ON.

03

JOINS Desnecessários

Inclusão de tabelas não utilizadas no resultado.





BORA FAZER UMA  
ATIVIDADE PARA  
FIXAR OS CONCEITOS!

MÃO NA  
MASSA



Exemplo...



# Avalie nossa aula de hoje na Adalove!



Aluno



Professor

MUITO OBRIGADO!



# Formações Acadêmicas

## Graduações

*Enfermagem*

*Ciência da Computação*

*Tecnologia em Redes de Computadores*

*Programação de Computadores*

*Sistema de Informação*

## Pós- Graduações

*Mestre em Engenharia Elétrica e da Computação*

*Doutor em Engenharia Elétrica e da Computação*

*MBA Em Data Center e Computação em Nuvem*

*MBA em Gestão da Tecnologia da Informação*

*MBA Engenharia de Software*



## Certificações

*DELL EMC PROFISSIONAL – Cloud Infrastructure and Services;*

*DELL PROVEN PROFISSIONAL – Information Storage and Management.*

*ISO 27001*

*MCSA Windows SERVER - 410, 411 e 412;*

*Linux – Lpic 1 e Lpic 2;*

*Microsoft Certified Professional;*

*Symantec Backup Exec;*

*Symantec Endpoint Protection,*

*Cobit;*

*ITIL;*

Professor

Dr. Cristiano

Benites



OBRIIGADO  
E LET'S BORA!