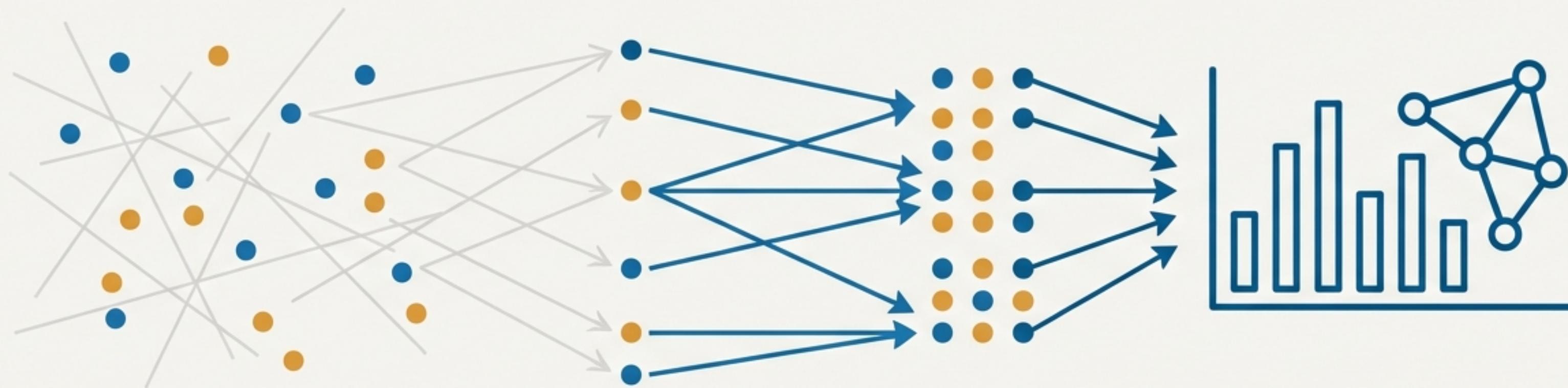


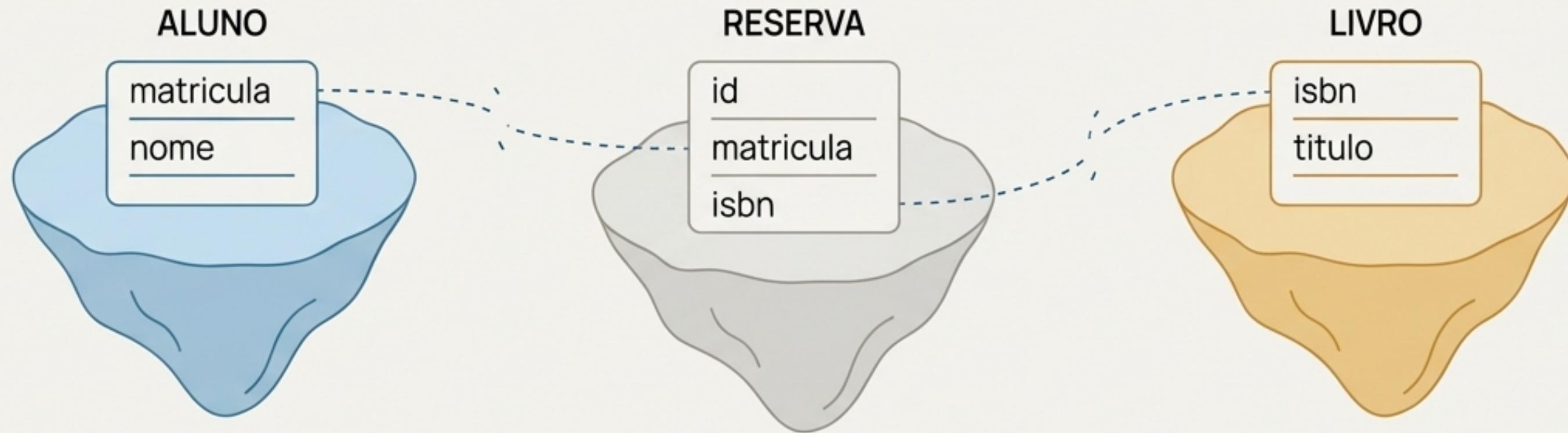
Extraindo Insights de Dados Relacionados

Um Guia Visual para Dominar `JOIN`, `GROUP BY` e `VIEWS` em SQL



Transforme tabelas separadas em relatórios poderosos.
Uma jornada do caos à clareza com as ferramentas essenciais do SQL.

O Desafio: Nossos Dados Estão em Ilhas Separadas



Temos os dados, mas as respostas estão escondidas.

Cada tabela contém uma parte da história:

- 'ALUNO': Quem são nossos usuários?
- 'LIVRO': Qual é o nosso acervo?
- 'RESERVA': Quem reservou o quê e quando?

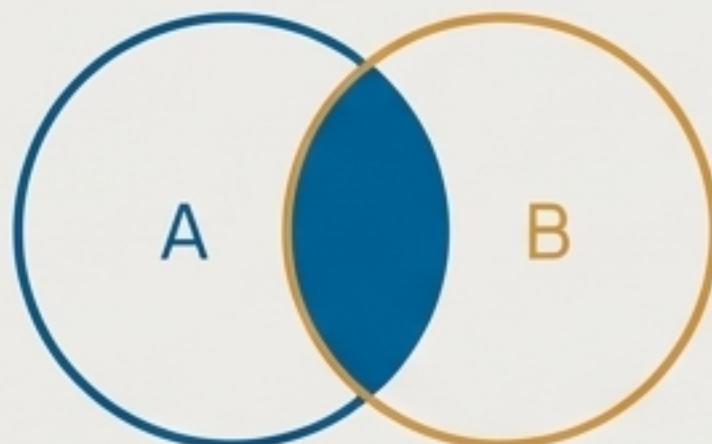
A Nossa Missão

Nosso objetivo é responder a perguntas críticas de negócio:

- ✓ Quais são os livros mais populares?
- ✓ Quais alunos nunca fizeram uma reserva?
- ✓ Quem são nossos leitores mais ativos?

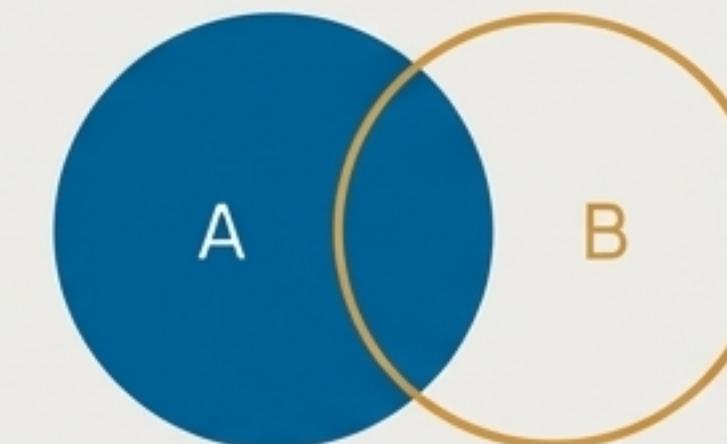
A Ferramenta Essencial: Construindo Pontes com `JOIN`

`JOIN` é a cláusula que nos permite combinar linhas de duas ou mais tabelas com base em uma coluna relacionada. É a primeira etapa para unificar nossa visão dos dados.



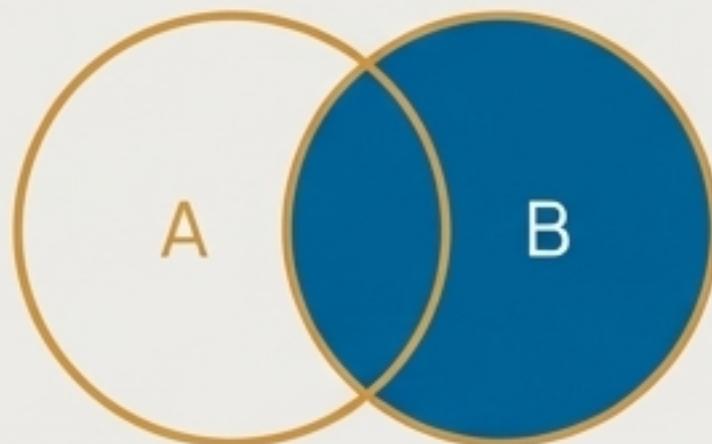
INNER JOIN

Apenas a interseção. Traz somente registros que têm correspondência em ambas as tabelas.



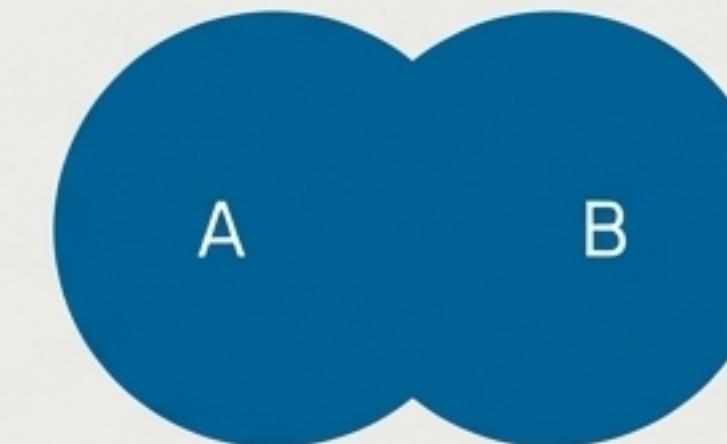
LEFT JOIN

Tudo da esquerda, mais as correspondências. Traz todos os registros da tabela A, preenchendo com NULOs onde não há par em B.



RIGHT JOIN

Tudo da direita, mais as correspondências. O inverso do LEFT JOIN.



FULL OUTER JOIN

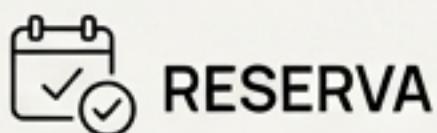
A união completa. Traz todos os registros de ambas as tabelas, com NULOs onde não há correspondências.

Lembre-se: `JOIN` combina colunas (alarga os dados), enquanto `UNION` empilha linhas (aumenta os dados).

'JOIN' em Ação: Quem Reservou Qual Livro?

Como podemos criar uma lista de reservas ativas que mostre o nome do livro em vez de apenas seu ISBN?

Antes do JOIN



RESERVA

id	matricula	isbn	status
{1,	'2021001'	'978-85-001'	'ativa'

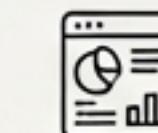


LIVRO

isbn	titulo
'978-85-001'	'Banco de Dados'

INNER JOIN

Depois do JOIN



Resultado

id	titulo	status
{1,	'Banco de Dados'	'ativa'

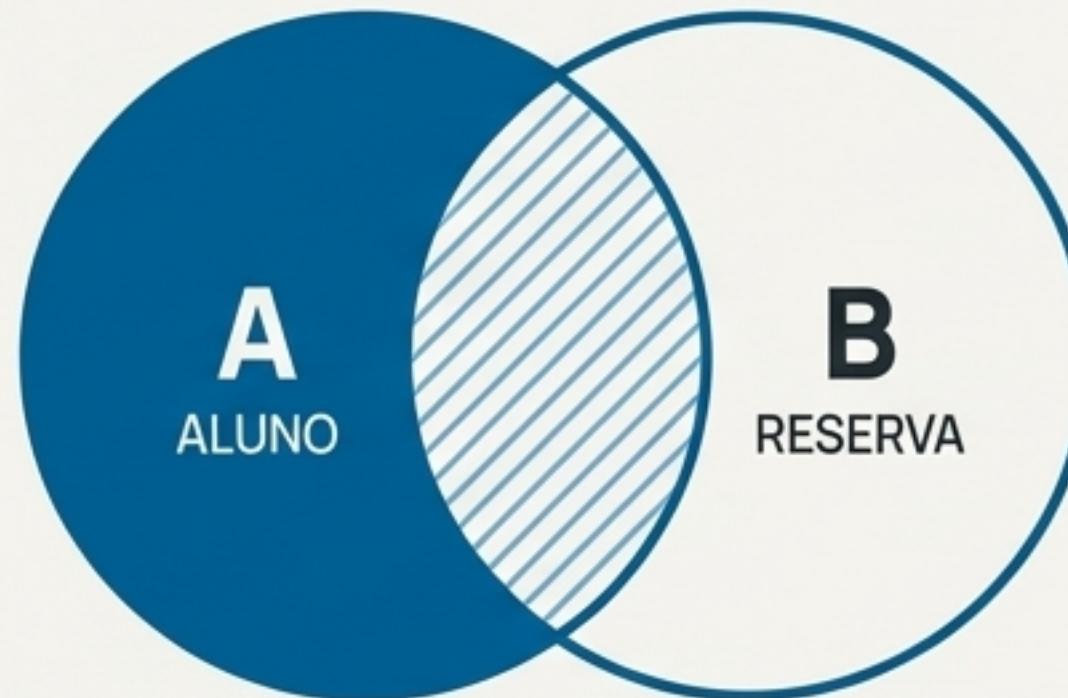
```
-- Listar reservas ativas com título do livro
SELECT
    R.id,
    L.titulo,
    R.data,
    R.status
FROM
    RESERVA AS R
INNER JOIN -- A palavra INNER é opcional
    LIVRO AS L ON L.isbn = R.isbn
WHERE
    R.status = 'ativa';
```

Com um JOIN, trocamos chaves estrangeiras (ISBN) por informações legíveis (título), tornando o relatório imediatamente útil.

A Descoberta do Ausente: O Padrão 'Anti-Join'

Tão importante quanto saber quem reservou livros é saber quem *nunca* reservou.
Como identificamos esses alunos?

LEFT ANTI-JOIN



Linhos de A sem
correspondência em B

A Lógica Explicada

1. Passo 1 ('LEFT JOIN'):

Começamos com um 'LEFT JOIN' da tabela 'ALUNO' para a 'RESERVA'. Isso garante que todos os alunos sejam listados, independentemente de terem ou não uma reserva.

2. Passo 2 ('WHERE IS NULL'):

Filtramos os resultados para manter apenas as linhas onde a chave da tabela 'RESERVA' é 'NULL'. Isso significa que o aluno não teve correspondência, ou seja, nunca fez uma reserva.

-- Alunos que nunca fizeram uma reserva

```
SELECT A.matricula, A.nome  
FROM  
    ALUNO AS A  
LEFT JOIN RESERVA AS R  
    ON A.matricula = R.matricula  
WHERE  
    R.id IS NULL;
```

-- A condição chave do anti-join

Encontrando Padrões: Agregando Dados com `GROUP BY`

Com os dados conectados, o próximo passo é resumir. As funções de agregação transformam milhares de linhas em métricas-chave. `GROUP BY` nos permite fazer isso para subgrupos específicos.



Relatório-Chave: Os Livros Mais Reservados da Biblioteca

Qual consulta nos daria um ranking dos livros por número total de reservas?

A Lógica da Consulta

- 1** **FROM e JOIN:** Conectamos LIVRO e RESERVA para associar títulos a reservas.
- 2** **GROUP BY:** Agrupamos todas as reservas pelo L.titulo.
- 3** **SELECT com COUNT:** Para cada grupo (título), contamos (COUNT) o número de reservas.
- 4** **HAVING:** Filtramos os *grupos*, removendo aqueles com poucas reservas (ex: HAVING COUNT(*) > 5). É diferente de WHERE, que filtra *linhas* antes de agrupar.
- 5** **ORDER BY:** Ordenamos o resultado para criar o ranking.

```
-- Livros mais reservados
SELECT
    L.titulo,
    COUNT(R.isbn) AS total_reservas
FROM
    LIVRO AS L
JOIN
    RESERVA AS R ON L.isbn = R.isbn
GROUP BY
    L.titulo
HAVING
    COUNT(R.isbn) >= 1 -- Filtra grupos
após a agregação
ORDER BY
    total_reservas DESC;
```

Consultas Aninhadas: A Precisão das Subconsultas

Uma **subconsulta** (ou subquery) é um SELECT dentro de outro SELECT. Ela permite realizar filtros complexos, onde a condição de uma consulta depende do resultado de outra.

Estrutura da Consulta & Exemplo no WHERE



Quando Usar

Quando precisar **filtrar dados** com base em um **resultado agregado** (ex: encontrar alunos que fizeram parte do grupo dos 'top 10% de reservas').

Para **comparações** com valores que **não são conhecidos** de antemão, mas que **podem ser calculados** (ex: livros mais caros que a média de preços).

Subconsulta em Ação: Encontrando Livros de "Alto Volume"

Como podemos listar os títulos dos livros que foram reservados 5 ou mais vezes (status "ativa" ou "atendida")?

A Lógica em Duas Etapas

1. A Subconsulta (Interna)

Primeiro, precisamos de uma lista de ISBNs que atendem ao critério de 'alto volume'. Esta consulta seleciona R.isbn da tabela RESERVA, agrupa por isbn e usa HAVING COUNT(*) >= 5 para encontrar os ISBNs populares.



2. A Consulta Externa

Agora, usamos essa lista para filtrar a tabela LIVRO. Esta consulta seleciona L.titulo de LIVRO onde o L.isbn está contido (IN) na lista gerada pela subconsulta.



```
SELECT L.titulo  
FROM LIVRO AS L  
WHERE L.isbn IN (  
    -- Subconsulta: retorna a lista de ISBNs  
    populares  
    SELECT R.isbn  
    FROM RESERVA AS R  
    WHERE R.status IN ('ativa', 'atendida')  
    GROUP BY R.isbn  
    HAVING COUNT(*) >= 5  
);
```

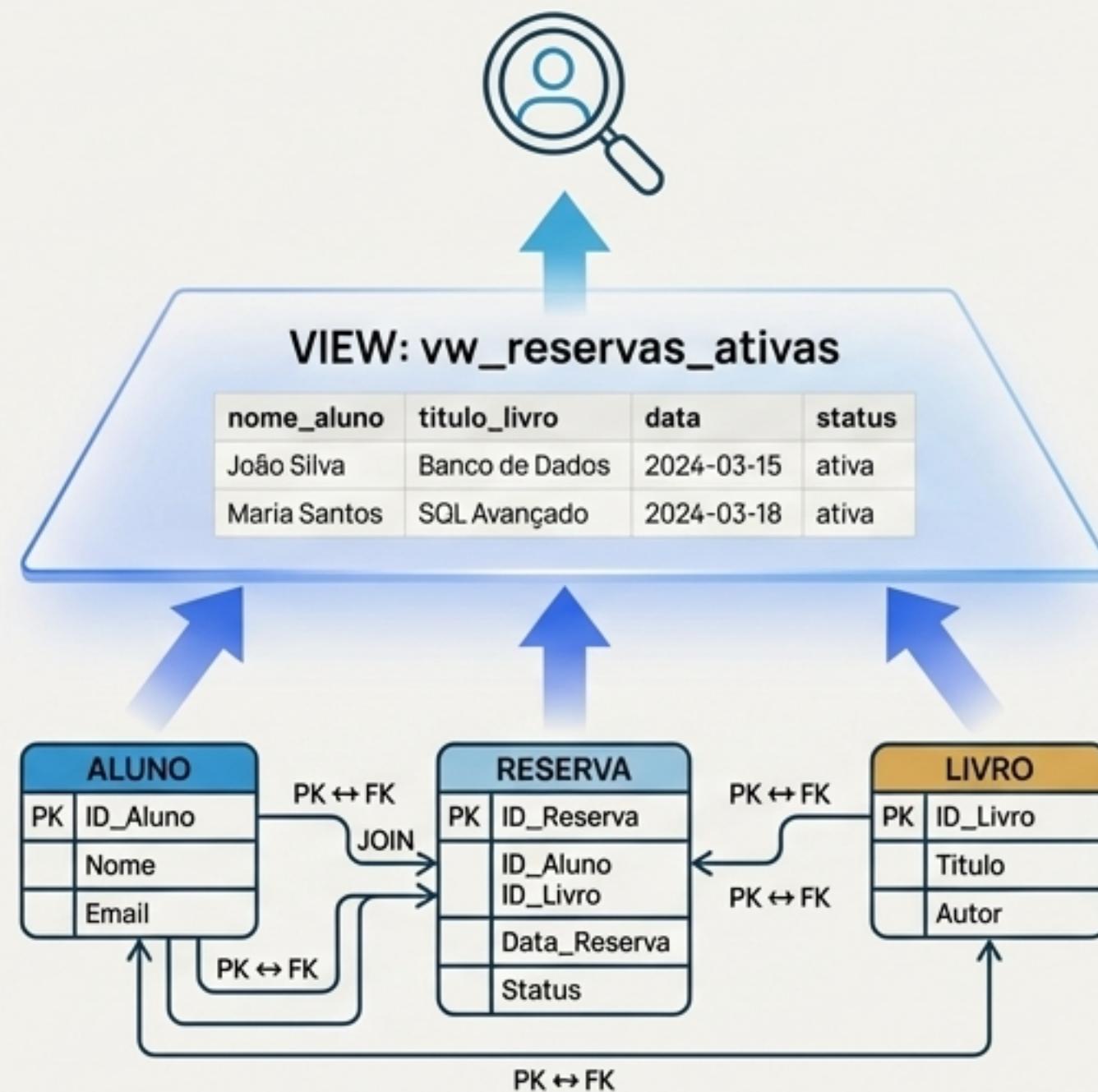
O Grand Finale da Clareza: Simplificando Tudo com VIEWS

E se pudéssemos salvar nossa consulta complexa e tratá-la como uma tabela simples? Isso é uma **VIEW**. Ela é uma tabela virtual baseada no resultado de um SELECT.

Consulta Simplificada

Camada de Abstração

Tabelas Base



Principais Vantagens



Simplificação: Esconde JOINs e lógicas complexas.



Reutilização: Evita reescrever as mesmas consultas repetidamente.



Segurança: Permite restringir o acesso a colunas ou linhas específicas.

Criando e Usando uma `VIEW`: `vw_reservas_ativas`

Passo 1: A Criação da `VIEW`

Primeiro, definimos a `VIEW` com a lógica de `JOIN` para criar um relatório legível de todas as reservas ativas.

```
CREATE VIEW vw_reservas_ativas AS
SELECT
    A.nome AS nome_aluno,
    L.titulo AS titulo_livro,
    R.data,
    R.status
FROM
    RESERVA AS R
    JOIN ALUNO AS A ON A.matricula = R.matricula
    JOIN LIVRO AS L ON L.isbn = R.isbn
WHERE
    R.status = 'ativa';
```



Passo 2: A Consulta Simplificada

Agora, qualquer pessoa pode consultar esta `VIEW` como se fosse uma tabela, sem precisar saber dos `JOINS` internos.

```
-- Contar quantas reservas ativas cada aluno
possui
SELECT
    nome_aluno,
    COUNT(*) AS qtde_reservas
FROM
    vw_reservas_ativas -- Simples e direto!
GROUP BY
    nome_aluno
ORDER BY
    qtde_reservas DESC;
```



Do Caos à Clareza: O Relatório Final da Biblioteca

Armados com `JOIN`, `GROUP BY`, e `HAVING`, agora podemos gerar os relatórios estratégicos que a biblioteca precisa.



Livros mais reservados?

```
-- Código 5.5
SELECT
    L.titulo,
    COUNT(R.isbn) AS total_reservas
FROM
    LIVRO AS L
JOIN
    RESERVA AS R ON L.isbn = R.isbn
GROUP BY
    L.titulo
ORDER BY
    total_reservas DESC;
```



Histórico de reservas por status?

```
-- Código 5.6
SELECT
    A.nome AS nome_aluno,
    L.titulo,
    R.data,
    R.status
FROM RESERVA AS R
JOIN ALUNO AS A ON A.matricula = R.matricula
JOIN LIVRO AS L ON L.isbn = R.isbn
WHERE R.status = 'ativa';
```



Empréstimos por aluno?

```
-- Código 5.7
SELECT
    A.nome AS nome_aluno,
    COUNT(E.id) AS total_emprestimos
FROM
    ALUNO AS A
JOIN
    EMPRESTIMO AS E ON E.matricula = A.matricula
GROUP BY
    A.nome
ORDER BY
    total_emprestimos DESC;
```

Cada uma dessas consultas, que antes parecia complexa, agora é uma ferramenta poderosa para a tomada de decisão.

Teste Seu Conhecimento: Aplicando os Conceitos em E-commerce

Imagine um banco de dados de e-commerce com as tabelas `CLIENTE(CPF, nome)` e `REGISTRO_VENDA(id, cliente_cpf, valor_total)`.



Desafio 1: Ticket Médio

Calcule o valor médio (`AVG`) de todas as vendas.

-- Código 5.10

```
SELECT AVG(valor_total) AS ticket_medio  
FROM REGISTRO_VENDA;
```



Desafio 2: Clientes Frequentes

Liste os clientes com mais de 5 compras, mostrando o total de pedidos.

-- Código 5.11

```
SELECT C.nome, C.CPF, COUNT(RV.id) AS total_pedidos  
FROM CLIENTE AS C  
JOIN REGISTRO_VENDA AS RV ON RV.cliente_cpf = C.CPF  
GROUP BY C.nome, C.CPF  
HAVING COUNT(RV.id) > 5  
ORDER BY total_pedidos DESC;
```

Seu Toolkit para Extrair Insights de Dados



JOIN

A ponte. Combina dados de múltiplas tabelas. 'INNER' para interseções, 'LEFT/RIGHT' para manter um dos lados.



Anti-Join

O detetive. Encontra o que *não* tem correspondência ('LEFT JOIN ... WHERE ... IS NULL').



GROUP BY

A lupa. Agrupa linhas para aplicar funções de agregação ('COUNT', 'SUM', 'AVG').



HAVING

O filtro de grupos. Aplica condições *após* a agregação.



Subconsulta

A ferramenta de precisão. Uma consulta aninhada para criar filtros avançados.



VIEW

A lente da clareza. Salva uma consulta complexa como uma tabela virtual para simplificar e reutilizar.

Com estas ferramentas, você está equipado para transformar dados brutos em inteligência acionável.