

A Jornada do Dado de Qualidade

Do SQL ao Dataset: Transformando **Dados Estruturados**
em Insights para **Machine Learning**



O Desafio Oculto do Machine Learning

Antes de qualquer algoritmo, existe um desafio que define o sucesso ou o fracasso de um projeto.



80%

do tempo de um cientista de dados é gasto limpando e preparando dados.



-40%

Um dataset mal estruturado pode gerar modelos com acurácia até 40% menor.



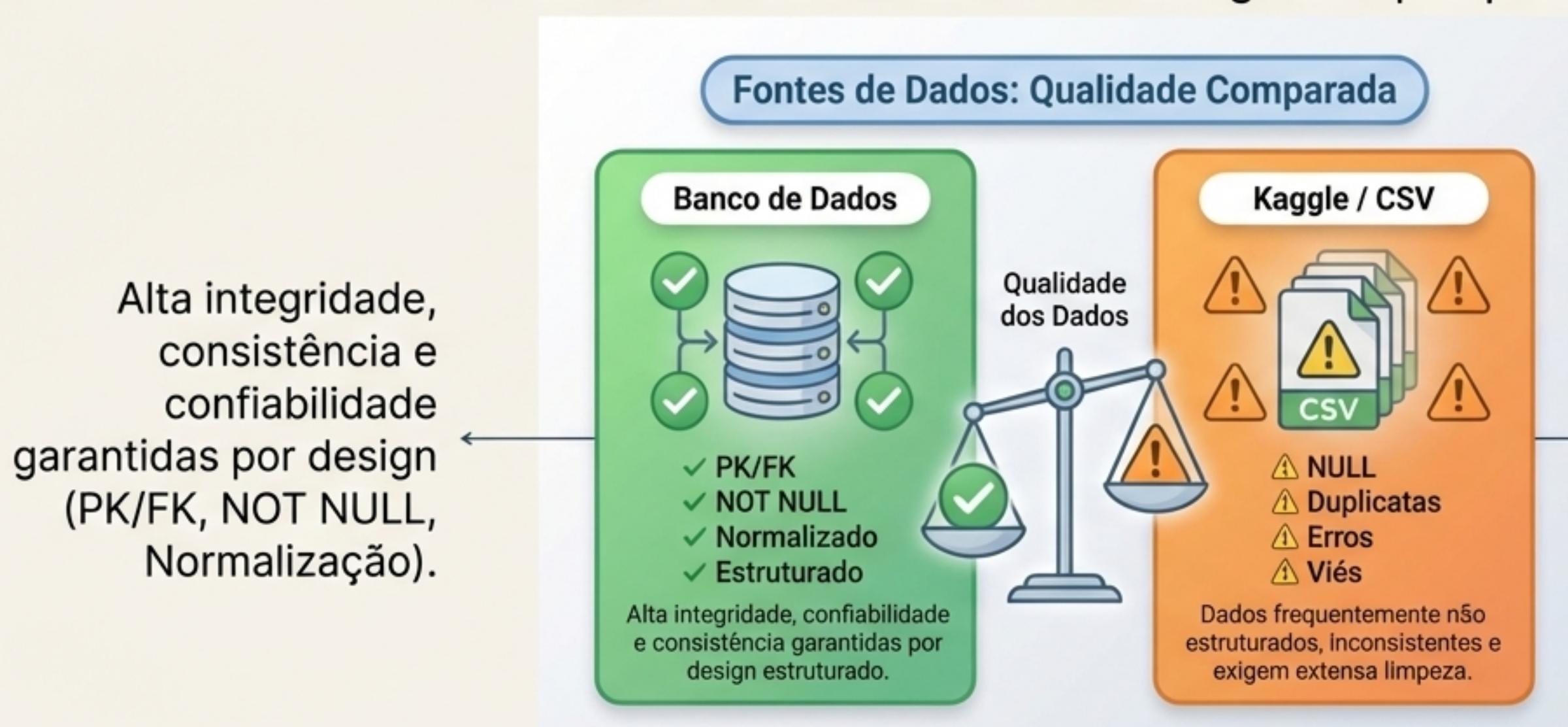
200.000+

O Kaggle, referência em datasets públicos, expõe a complexidade: mais de 200.000 datasets que frequentemente exigem limpeza extensa.

Fonte: Forbes/CrowdFlower 2016, Kaggle 2024, Boas práticas de ML.

A Resposta Começa na Fonte: Banco de Dados vs. CSVs

A qualidade do seu modelo de ML não começa no Python, mas na origem dos seus dados.
A escolha da fonte é a decisão mais estratégica no pré-processamento.



Escolher a fonte correta é crucial para análises precisas e modelos de IA confiáveis.

Mapeando a Jornada: O Framework CRISP-DM

Projetos de dados de sucesso seguem um roteiro. O **CRISP-DM** é a metodologia padrão da indústria, e nosso foco estará nas fases de entendimento e preparação dos dados.

Nosso foco: transformar o entendimento dos dados em um dataset pronto para a modelagem.

1. Business Understanding

Definir metas e métricas de sucesso.

2. Data Understanding

Coletar, descrever e explorar dados.



3. Data Preparation

Limpar e transformar dados.



4. Modeling

Selecionar algoritmos e ajustar hiperparâmetros.

5. Evaluation

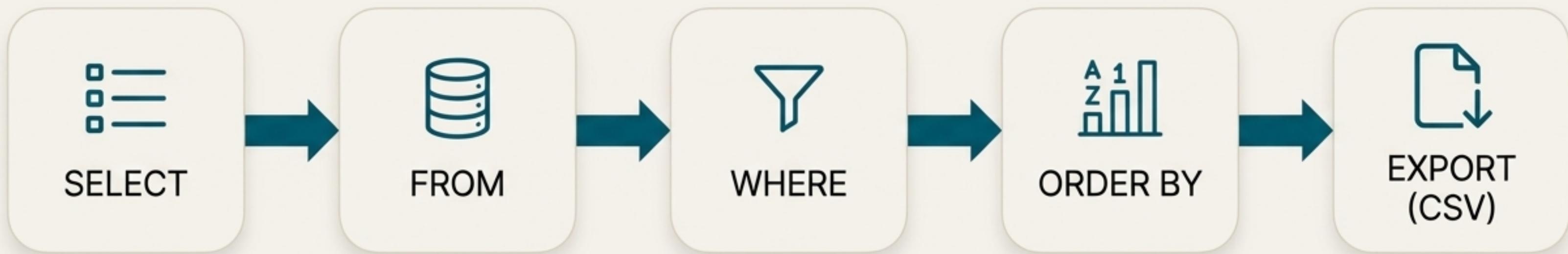
Avaliar resultados e riscos.

6. Deployment

Implantar e monitorar.

Parte 1: A Fonte – Extraindo Valor do Banco de Dados

Um banco de dados relacional não é apenas um repositório; é a primeira linha de defesa da qualidade dos dados. Ele concentra informações estruturadas, consistentes e já validadas.



Com **JOIN**, agregações (**GROUP BY**) e filtros (**WHERE**), nós não apenas extraímos dados, mas também começamos a modelar o dataset diretamente na consulta, garantindo coesão e relevância.

O Pipeline Visual: Do SQL ao Dataset

Este é o fluxo que transforma a lógica relacional em um arquivo plano e universal, pronto para qualquer ferramenta de análise.



Parte 2: A Ponte – Do Banco de Dados ao Modelo de ML

O que é um 'Dataset' no contexto de Machine Learning?

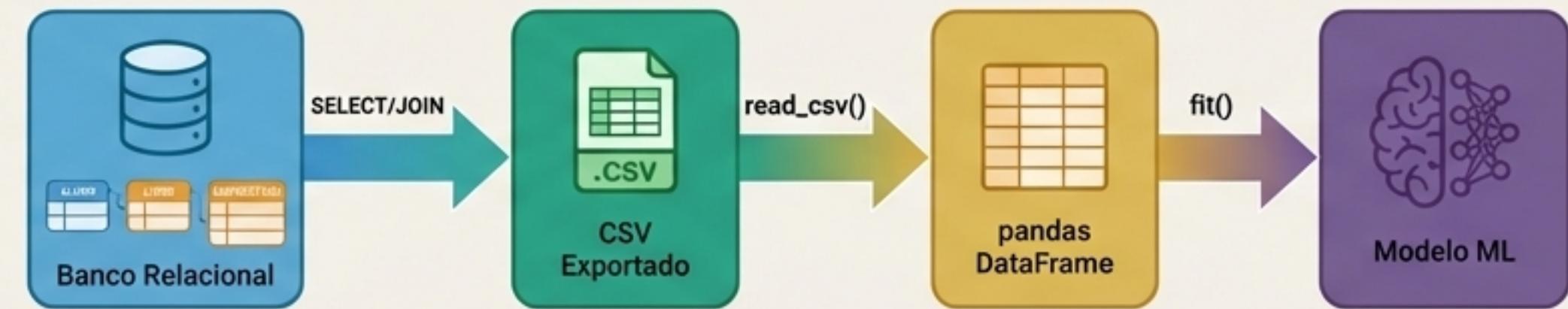
É o encontro entre o **modelo de dados (BD)** e o **modelo de aprendizado (ML)**. Um dataset bem-sucedido traduz a lógica relacional e normalizada para o formato vetorial que os algoritmos de ML compreendem.



O caminho vai do dado validado no banco até a matriz de features que alimenta o modelo. Cada etapa adiciona valor e garante consistência.

Visualizando a Ponte: O Caminho Completo

1. **Banco Relacional:** Começamos com tabelas normalizadas (Aluno, Livro, Empréstimo).
2. **SELECT/JOIN:** Unificamos as tabelas em uma única visão lógica.
3. **.CSV Exportado:** Materializamos essa visão em um arquivo.
4. **pandas DataFrame:** Carregamos os dados na memória com `read_csv()` para manipulação.
5. **Modelo ML:** Usamos o DataFrame limpo para treinar o modelo com `fit()`.



Ponto Crítico: `JOINs` e colunas derivadas devem ser feitos na fase de SQL para garantir consistência antes da vetorização das features.

O Valor de uma Fundação Sólida

Cada princípio de modelagem de dados no BD tem um impacto direto na qualidade das features do seu modelo de ML.

Etapa (no BD)	Objetivo	Benefício no ML
Normalização	Reducir redundância e anomalias.	Consistência e dados confiáveis. Evita vazamento de dados (data leakage) e features correlacionadas.
Recomposição com JOINs	Criar uma visão integrada do negócio.	Features coerentes e com significado contextual.
Derivação de Colunas	Criar colunas com agregações ou cálculos.	Engenharia de features na fonte, criando indicadores e sinais mais fortes para o modelo.

Parte 3: A Lapidação – Pré-processamento e Limpeza

Muitas tarefas de 'limpeza de dados' são, na verdade, sintomas de uma modelagem de dados deficiente. Uma boa modelagem é o melhor pré-processamento.

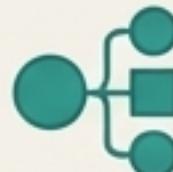
Resolvido no Banco de Dados (A Prevenção)



Atributos Não Atômicos: Resolvidos com a 1^a Forma Normal (1FN).



Atributos Não Identidade.

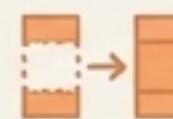


Redundância e Anomalias: Resolvidas com a 2^a (2FN) e 3^a (3FN) Formas Normais.



Valores Nulos Críticos: Minimizados com a restrição `NOT NULL`.

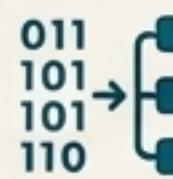
Tratado na Análise (A Correção)



Valores Nulos Estratégicos: Imputação ou descarte.



Outliers: Detecção e tratamento (recorte, transformação).

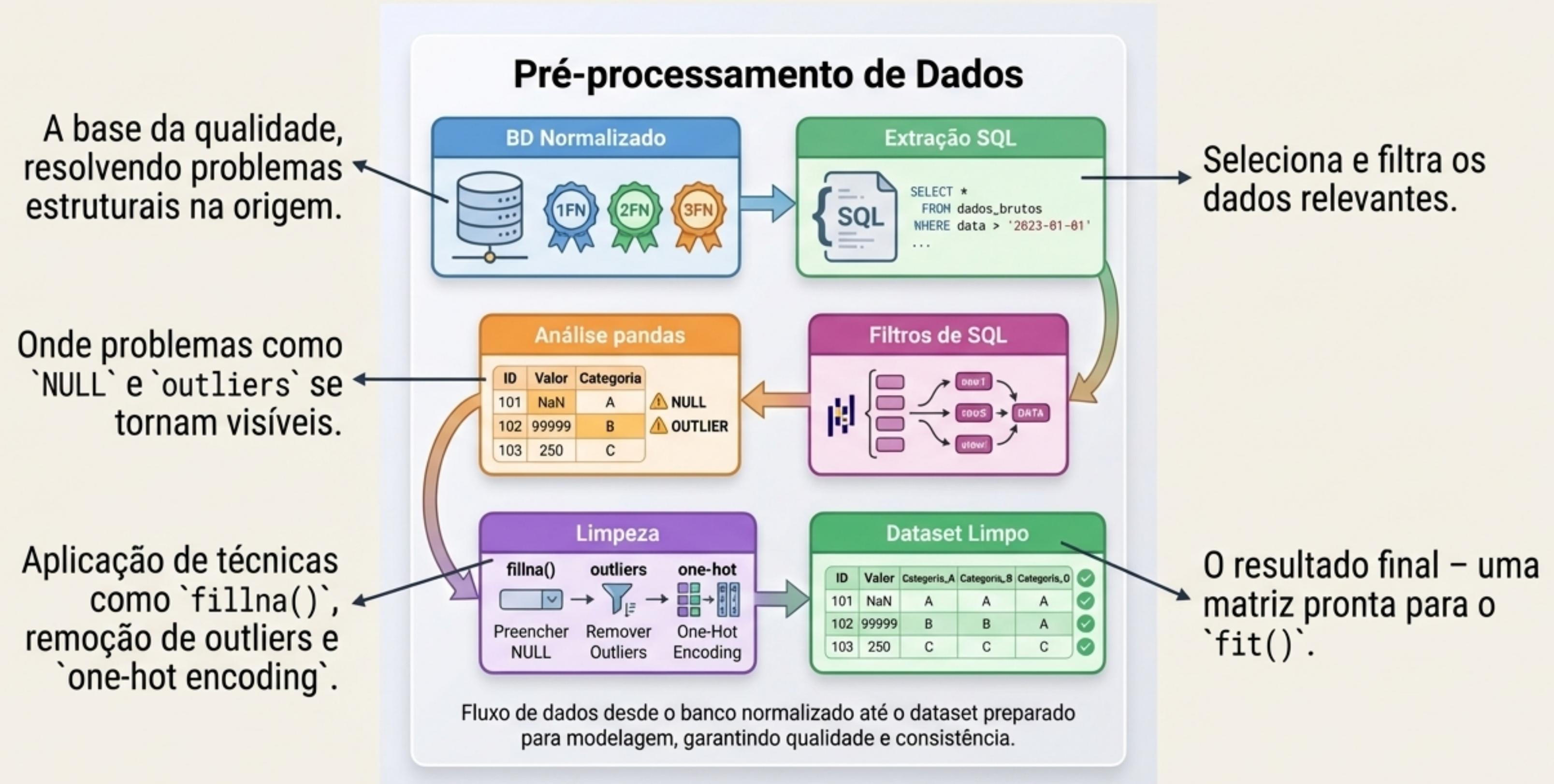


Codificação (Encoding): Conversão de categorias (One-Hot, Target).



Normalização/Escalonamento: Ajuste de escala de variáveis (Min-Max, Z-score).

O Fluxo de Trabalho do Pré-processamento



Estudo de Caso: Geração de um Dataset de Empréstimos (Biblioteca)

Contexto:

Nosso objetivo é gerar um dataset **consolidado** para analisar o comportamento de **emprestimos** e, potencialmente, prever o risco de **atraso** na devolução de livros.

As Fontes de Dados:

- Tabela `ALUNO` (Informações do estudante)
- Tabela `LIVRO` (Informações da obra)
- Tabela `EMPRESTIMO` (A tabela de fatos que conecta alunos e livros)

O Processo:

1. Unificar as três tabelas usando `JOINs` por chaves primárias e estrangeiras.
2. Criar colunas derivadas úteis (ex: dias de atraso).
3. Exportar o resultado consolidado para um arquivo CSV.



Construindo o Dataset com SQL: A Consulta de Extração

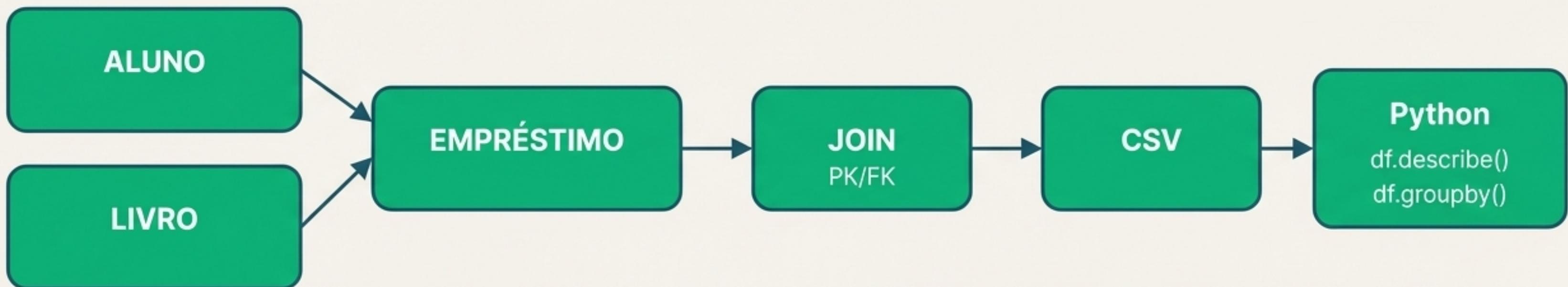
Unificação das
tabelas.

```
SELECT
    A.Matricula,
    A.Nome AS Nome_Aluno, ←
    L.Titulo AS Titulo_Livro,
    E.Data_Emprestimo,
    E.Data_Devolucao_Real,
    -- Coluna Derivada: Engenharia de Feature na Fonte
    (E.Data_Devolucao_Real - E.Data_Devolucao_Prevista) AS Dias_Atraso
FROM
    ALUNO AS A
INNER JOIN
    EMPRESTIMO AS E ON A.Matricula = E.Matricula_Aluno
INNER JOIN
    LIVRO AS L ON E.Codigo_Livro = L.Codigo
ORDER BY
    A.Nome, E.Data_Emprestimo;
```

Uso de aliases
para clareza.

Engenharia de features: O SQL já está
criando a feature 'Dias_Atraso'.

Do Conceito à Prática: O Fluxo do Estudo de Caso



Explicação do Fluxo

1. **Origem:** As tabelas `ALUNO` e `LIVRO` alimentam a tabela `EMPRESTIMO`.
2. **Unificação:** `JOIN` via PK/FK combina tudo em uma visão única.
3. **Exportação:** O resultado é salvo como `CSV`.
4. **Análise:** Em Python, o CSV é carregado e explorado com `pandas`.

Próximos Passos em Python

Após gerar o CSV, o próximo passo seria a análise exploratória: `df.describe()`, `df.groupby('Titulo_Livro')[['Dias_Atraso']].mean()`, e visualizações como `sns.histplot(df['Dias_Atraso'])`.

A Receita para a Qualidade: Onde o Bom ML Realmente Começa

A qualidade do seu modelo de Machine Learning é um reflexo direto da qualidade da modelagem do seu banco de dados.



Comece Certo: Use um banco de dados normalizado como sua "fonte da verdade" para garantir integridade e consistência.



Extraia com Inteligência: Use o poder do SQL (`JOINs`, agregações, derivações) para fazer a maior parte da estruturação e engenharia de features.



Lapide com Precisão: Utilize Python e `pandas` para as etapas finais de limpeza e pré-processamento que não podem ser resolvidas na origem (outliers, encoding, scaling).

